# DECENTRALIZED OPTIMIZATION AND ANALYTICS FOR LARGE SCALE POWER SYSTEM PROBLEMS

A Dissertation
Presented to
The Academic Faculty

By

Paritosh Parnad Ramanan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering
College of Engineering

Georgia Institute of Technology

December 2020

**DECENTRALIZED OPTIMIZATION AND ANALYTICS FOR LARGE SCALE POWER SYSTEM PROBLEMS**

Thesis committee:


Dr. Nagi Gebraeel, Advisor
School of Industrial and Systems Engineering
*Georgia Institute of Technology*

Dr. Santanu Dey
School of Industrial and Systems Engineering
*Georgia Institute of Technology*

Dr. Edmond Chow, Co-Advisor
School of Computational Science and Engineering
*Georgia Institute of Technology*

Dr. Andy Sun
School of Industrial and Systems Engineering
*Georgia Institute of Technology*

Dr. Michael Rabbat
Facebook AI Research
*Montreal*



Date approved: October 8, 2020

To my prayers, my parents, teachers and friends. Without whom this research would not have been possible.

## ACKNOWLEDGMENTS

I cannot even begin to express my humble gratitude to my advisors Prof. Nagi Gebraeel and Prof. Edmond Chow. Their sage advice and constant support was critical in making my research possible. The insights and guidance that they bestowed on me during my PhD has given me a lifetime's worth of knowledge that I will forever cherish and for that I shall be forever grateful to them. Thank you both from the bottom of my heart for making my PhD so rewarding and fulfilling!

My thesis would not have been possible without the support and mentorship of Murat Yildirim. I still remember all the late nights and weekends we spent in the early years. Little did I know that the subject of the paper I was trying to push through back then was complex enough to lead to a whole PhD thesis. Thank you for correcting, mentoring and bearing with me for all these years!

Thank you to my parents, especially my Mom who would always bear the brunt of my paper rejections, fears and apprehensions I had about my future. Its thanks to her and her prayers that I have reached this position. Thanks to my Dad who bore with my technical rants patiently and always eagerly listened to my research ideas and progress updates! Thanks also to my sister whose constant admiration for my career in research, made it all the more worth it. Her uncannily sage and timely advice and patient listening got me out of some dark times!

Special thanks to Rajeev uncle and Sapna aunty. Ever since I have landed in the US, they have been a constant pillar of support for me. I express my heartfelt gratitude to them for all their help! Thank you uncle and aunty!

I also have a deep sense of gratitude for my friends Aishwarya, Kishen and Ganesh. Aishwarya's baking and Kishen's leg pulling gave me a lot of cheer on otherwise gloomy days. Thank you to Ganesh for always entertaining me with trivia and partaking with me on some crazy adventures! Also a shout out to all the crazy late night drives we used to

have on weekends, those days would be missed!

My friends from undergrad, Akash and Pavan deserve a special mention. Akash, who somehow put up with me and my music for an inordinate amount of time and was gracious enough to lend his couch for me to crash on so many times and took countless road trips with me, thanks man! Pavan, my friend from undergrad and roommate for the greater part of the PhD has been directly impacted by my absent mindedness on account of me being preoccupied by research countless times! I apologize and cant thank you enough for bearing with my stupidities for a ridiculous amount of time! Thank you guys for being with me always!

I am also grateful to my friends Kirthana and Emma. A chance encounter with them during Integer Programming led to such deep friendship for life. Thanks for dropping over so many times to my place and being the guinea pigs for my world awesome Indian cooking! I fondly remember and recollect all our times together. Thanks for being there!

Deep gratitude also to my girlfriend Prerna. She has somehow managed to tolerate me for so long and for that I have deep respect for her. Thank you for encouraging me, listening to my rants and always being there for me!

I'd like to thank Siva and Erik at Sandia National Labs for hosting me for a wonderful summer in 2017 which was super enjoyable and in which I learnt a lot! Thank you to Kiyoshi at NEC Research Labs for mentoring me during the summer of 2019! I also have deep respect to the Sam Nunn Fellowship which introduced me to some amazing people like Linda and gave me an in-depth experience of the world of government and research.

I have the deepest regards for my alma mater, Georgia Tech for providing me so many opportunities to acquire knowledge in such diverse aspects of life that I never imagined I'd acquire someday! My previous alma maters Georgia State and BITS-Pilani deserve special mention. Had it not been for the wonderful, encouraging environment at BITS-Pilani, I wouldnt have ever ventured into the world of research. Had Georgia State not fostered my love for research I wouldnt be here at all! Thanks to all these wonderful institutions.

Lastly, thanks to those people who never believed in me nor had any faith in my abilities. Had it not been for their constant barbs, I'd never have found the burning desire to pull through!

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## SUMMARY

Large scale power networks form the backbone of the global energy infrastructure. Power system optimization problems are geared towards large scale planning problems in power systems. The solutions to these problems offer better utilization of system resources and therefore such problems form a significant part of power systems research. However, large scale planning and optimization problems demand efficient computational schemes that respect the data privacy of asset owners and operators as well.

Decentralized methods have lately emerged as a means to tackle the different operational issues like data privacy and computational efficiency. Decentralized methods localize the problem and data component in a multi agent system like the power grid. Therefore, decentralized approaches towards planning problems in power systems could be an attractive way for utilities to derive globally optimal solutions without divulging their local data in a computationally efficient fashion. As a result, decentralized computational paradigms for large scale planning problems in power systems are gaining popularity.

In this thesis, we explore novel ways to solve computationally challenging planning and analytics problems in a decentralized manner using synchronous as well as asynchronous computational models. We specifically focus on decentralized formulations of unit commitment, joint operations and maintenance, differential privacy based unit commitment and maintenance as well as a blockchain based, decentralized analytics methodology for replay attack detection.

# CHAPTER 1
# OPTIMIZATION PROBLEMS IN POWER SYSTEMS

## 1.1 Introduction

Large scale power networks form the backbone of the global energy infrastructure. A surge in global power consumption now demands better energy distribution schemes as well as increased efficiency in the management of power network resources and assets. Power network optimization problems aim at better utilization of system resources and form a significant part of power systems research.

A large-scale power network is topologically divided into a number of regions which may represent different power utility companies or subsidiaries. Power utility companies are often part of a framework known as a Regional Transmission Operator (RTO) which facilitates trading of power with dynamic pricing among its various member companies [1]. RTOs set up a mutually coordinated market place which influences prices by inducing a desirable system level behavior.

As part of an RTO, a power company must make decisions on optimum production, commitment and maintenance schedules. These decisions are in turn influenced by system level behavior and dynamic operations and maintenance costs. Further, a comprehensive planning framework employed by a utility company takes into account production and commitment decisions denoted by the operations problem and determining maintenance schedules denoted by the maintenance problem. Therefore, such an extensive optimization scheme couples the operations part with the maintenance problem, leading to a computationally challenging problem to solve. In addition, owing to the binary nature of decisions in such types of planning problems, the resulting optimization formulation ends up being Mixed Integer in nature.

Traditionally, a centralized model has been employed to solve power network planning problems. However, the centralized model has several drawbacks. First, centralized methods are unable to isolate potentially sensitive commercial operations data of the various power companies or regions. Second, the performance of a centralized model deteriorates with increase in network size leading to poor scalability thereby making it unsuitable for large-scale power networks. Third, an attack on a single node can compromise the entire power network.

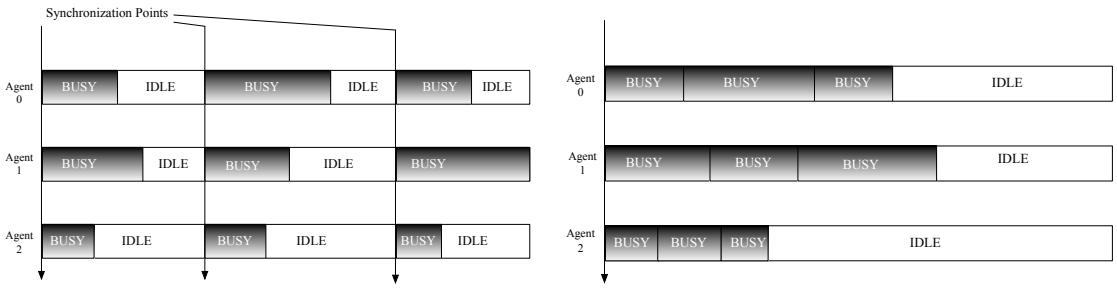
Decentralized optimization methods have lately emerged as a means to tackle the different operational issues presented above. Owing to a loose coupling between regions, the global optimization problem can be decomposed into smaller subproblems, each corresponding to a particular regional power subsidiary. By iteratively solving each region's subproblem locally and exchanging information with neighboring subproblems, we can completely decentralize the solution to large scale planning problems. Since each subproblem is locally held by the region itself, decentralized methods retain privacy of commercial data pertaining to each region. Further, decentralized methods enable solving for the global optimum only on the basis of local infrastructural data and relevant operational data points of neighboring regions thereby improving scalability.

### 1.1.1 Models of Computation: Asynchronous and Synchronous

In this work we focus on the asynchronous and synchronous models of computation as a means to obtain computationally efficient solutions.

An asynchronous approach involves all agents solving their local subproblems with the latest available information. On the other hand a synchronous approach involves iterations wherein an agent must wait for all other agents to finish their local computation before moving to the next iteration.

There are numerous benefits of using an asynchronous computation model. First, it is more amenable to a heterogenous computing environment, wherein all agents need not




(a) Synchronous (b) Asynchronous

Figure 1.1: Illustration of Synchronous and Asynchronous communication styles

necessarily have the same hardware. In such a case, an asynchronous model is able to deliver superior computational performance. Second, even with good hardware, there may be a significant imbalance in the local problem sizes of each agent which is handled elegantly by the asynchronous model. Lastly, in a geographically distributed computational setup, asynchronous methods offer robust solutions in the event of a cyber attack or hardware failures .

We develop a decentralized solution framework to the two important planning problems of operations and maintenance. Our approach to solving operations problem involves solving the Unit Commitment problem for the day ahead planning horizon. We then formulate the maintenance problem that relies on efficient solutions to the UC problem in order to determine optimum generator maintenance schedules based on sensor data.

### 1.1.2 Unit Commitment Problem

Unit Commitment (UC) problem in power networks is a key power systems planning problem that determines the optimal power generation schedule for a fleet of networked generators. The UC problem is solved multiple times during the day by power network utility companies for robustness purposes. In addition, the UC problem is a building block for various planning problems in power systems including sensor driven maintenance and transmission switching. Therefore, computationally scalable, efficient solution frameworks for the UC problem will have a direct impact on solution times of various other planning

problems as well. UC is a computationally challenging problem due to its scale, discrete nature and the vast amount of data that is essential to obtain a solution.

A regional decomposition strategy allows us the flexibility to choose an asynchronous model of computation wherein regions can progress with their local computation on availability of fresh information from any of their peers. This removes computational bottlenecks and improves computational efficiency. Therefore, the first part of this work focuses on an asynchronous decentralized problem formulation of the UC problem. Our asynchronous method provides faster convergence with superior solution quality rivaling the centralized methods.

### 1.1.3 Sensor Driven Generator Maintenance

The emergence of Cyber Physical Systems (CPS) has yielded unprecedented opportunities for close scrutiny of large scale power systems. In recent times, such systems have also come to provide valuable insights and knowledge on important operational and reliability parameters in power systems [2]. Sensor driven techniques have recently started gaining momentum within the field of power systems as a means to intelligently solve problems of sensor-driven maintenance, and operations [3]. Decreasing hardware costs have resulted in an even greater number of sensors being used to monitor power network equipment, in turn leading to a deluge of data being collected by these sensors. Advances in computational and communication capabilities of these sensors now enable us to collect rich real-time data from large number of assets in power systems[4].

Sensor data is difficult to process due to the size and scale of most modern power systems. Power utility companies solve for optimal operations and maintenance schedules in order to operate their network with maximum efficiency. Sensor data can be effectively used to coordinate operations with optimal maintenance schedules based on predicted remaining lifetimes for every generator. This avoids premature maintenance of generators thereby extracting the most from their remaining lifetime while minimizing the risk asso-

ciated with catastrophic failures. Integrating operations into maintenance scheduling introduces another layer of complexity. Maintenance and operations in power systems is highly interdependent: generator failure or maintenance prevents the generators from producing power. Therefore, it is crucial to mitigate the negative impacts of maintenance on operations through a formulation of a joint maintenance and operations optimization model.

### 1.1.4 Privacy Preserving Mixed Integer Decentralized Planning

As decentralized methods for solving planning problems become popular, privacy concerns regarding the operational characteristics of the local subproblems also need to be addressed. Such privacy concerns arise because decentralized formulations of planning problems require sharing of consensus quantities of flow and phase angles between regional subproblems. Using consensus quantities, it is possible to estimate the operational state of the local subproblem itself, thereby requiring algoeithms with strong privacy guarantees. Usually, these types of sharing must happen over public facing communication channels, which amplifies privacy concerns.

Differential privacy (DP) [5] has been touted as a solution to address privacy concerns arising from public release of sensitive data sets. The DP paradigm operates by injecting a well calibrated noise into these data sets guaranteeing that the true values can never be determined. However, the application of DP for protecting consensus quantities in decentralized formulations of planning problems largely remain understated. Ensuring differential privacy guarantees for decentralized planning problems is a challenging task. First, DP perturbations to consensus quantities must be engineered such that solution quality for the decentralized mixed integer formulations rivals that of their centralized counterparts. Moreover, on account of DP based noise, the convergence characteristics of these algorithms must suffer. Therefore, it is critical to develop privacy preserving decentralized formulations for planning problems which are capable of delivering good quality results with reasonable convergence expectations.

### 1.1.5 Blockchain based Decentralized Analytics for Data Privacy

Large scale power systems are comprised of regional utilities with IIoT enabled assets that stream sensor readings in real time. Such large scale power systems are actually interconnected at the physical level due to reliability constraints mandated by law. At an organizational level, these power systems are managed by multiple utilities which lack an efficient mechanism for sharing real-time cyber analytics insights. The alternative for detecting a cyber attack in real-time is to migrate data from utilities to a central aggregator or server which can estimate the cyber health of the entire large-scale system. However, such a system is infeasible due to a number of reasons. First, utilities are hesitant to share real-time data with a central coordinator owing to the business sensitivity of these data points. Second, there needs to be a significant investment required to setup a trusted third party that will take care of the global aggregation of insights.

Due to the recent advancements in blockchain, specifically with the onset of Smart Contracts (SCs), the process of decentralized aggregation among multiple distrusting parties has become significantly simpler. The concept of blockchain can therefore be effectively applied to compute a globalized, cyber health status of a large scale power system without transferring any sensor data. Further, by leveraging SCs orchestrated on a private blockchain, whose members comprises of the utilities themselves, data privacy is maintained, compute is decentralized and the requirement for a central aggregator is eliminated.

### 1.1.6 Organization

The rest of this thesis is organized as follows. In Chapter 2, we establish a synchronous solution mechanism to the decentralized sensor driven optimization problem that provides a comparable solution to the centralized method. In Chapter 3, we tackle the more fundamental decentralized UC (DUC) problem and explore an asynchronous solution for its computational benefit. We show that the asynchronous approach presented in Chapter 3 potentially leads to faster solution times showing good computational promise.

In Chapter 4, we strengthen our solution to the DUC problem with the help of privacy preserving valid inequalities that are computed asynchronously leading to a significant improvement in solution quality. We introduce an interleaved binary approach to our asynchronous method that improves convergence times for the asynchronous solution drastically.

In Chapter 5, we present our work that is driven by a subgradient based solution to the integrated sensor driven maintenance and UC problem for planning horizons spanning many months. In this approach, by dualizing the maintenance limit constraint that is coupled across weeks, we parallelize weekly operations locally with the help of the subgradient method. We orchestrate multithreading for every subproblem while employing the distributed memory paradigm to communicate among regions.

In Chapter 6, we propose a differential privacy driven approach geared towards decentralized formulations of mixed integer operations and maintenance optimization problems that protects network flow estimates. We prove strong privacy guarantees by leveraging the linear relationship between the phase angles and the flow. To address the challenges associated with the mixed integer and dynamic nature of the problem, we introduce an exponential moving average based consensus mechanism to enhance convergence, coupled with a control chart based convergence criteria to improve stability.

In Chapter 7, we propose a blockchain based decentralized framework for detecting coordinated replay attacks with full data privacy. We develop a Bayesian inference mechanism employing locally reported attack probabilities that is tailor made for a blockchain framework. We compare our framework to a traditional decentralized algorithm based on the broadcast gossip framework both theoretically as well as empirically. With the help of experiments on a private Ethereum blockchain, we show that our approach achieves good detection quality and significantly outperforms gossip driven approaches in terms of accuracy, timeliness and scalability.

# CHAPTER 2

# DECENTRALIZED SENSOR DRIVEN MAINTENANCE

## 2.1 Introduction

Due to increasing performance requirements coupled with lack of investment in power infrastructure, efficient ways for monitoring condition of equipment and preventing their failures is becoming more critical. Generators, which form the backbone of any power network, are complex machines with thousands of moving parts which are critical to their reliable operation. It becomes imperative that a fault in a generator be immediately identified and rectified. Failure to do so may prove catastrophic causing massive generator downtime, immense economic and energy losses and structural damages to the grid. A well-timed maintenance schedule is critical to ensure that the generator fleet keeps operating at its highest efficiency without any failure. Maintenance of a generator is carried out to prevent catastrophic failures during operation and involves taking the generator out of service for a fixed period of time before being put back into service. Thus, an important question in power networks is to determine a cost-effective way of performing maintenance on a fleet of generators, while meeting the power demand imposed by the operational requirements of the network.

An optimal sensor driven maintenance and operations schedule goes a long way in ensuring profitability and significant savings for utilities [6]. Using sensor data for predictive maintenance however comes at a cost. Solving the operations and sensor driven maintenance problem is a computationally intensive task owing to its Mixed Integer nature. A centralized model is hampered by computational scalability owing to increased communication latency with growing network and fleet size. Further, sensor data is very critical to the business interests of utilities and OEMs and may be subject to data residency re-

quirements as well. A centralized repository of sensor data streamed from all assets of participating utilities might lead to irreparable damage in case of a data breach. This makes it impractical to transfer and store streaming sensor data in a centralized location.

In order to address these issues, decentralized methods have been proposed as a means to achieve high scalability and also to handle dynamic nature of streaming data. Decentralized methods in power networks are being spearheaded by recent advancements in computing that have led to an increase in computational speed to an order of magnitude faster than communication speed. As processors become more powerful and less expensive, it now becomes computationally more efficient to engage multiple such processors in a decentralized manner to ensure highly scalable systems [7].

Our contributions in this chapter are as follows:

- We develop a privacy preserving decentralized sensor driven maintenance framework that relies on the decomposition of the power network topology into multiple independent regions. Our framework is capable of leveraging locally available regional sensor data to reach a globally optimal maintenance solution in a decentralized fashion.

- We provide an iterative ADMM based consensus algorithm to power our decentralized sensor driven maintenance framework that achieves flow and phase angle balance between regions. The proposed communication architecture reaches to globally optimal maintenance and operational schedules only by exchanging corresponding network flow values while keeping the streaming sensor data from regional assets and regional infrastructural information private.

- Locally, our framework employs a sensor based condition monitoring scheme that uses a Bayesian updation scheme to estimate the residual lifetime of every generator. Our problem formulation leverages the residual lifetime calculation to make an informed maintenance decision with due regards to long term operational planning as

well.

- We provide an experimental High Performance Computing(HPC) driven platform that enables us to perform rigorous testing of our framework. Our platform is based on using the Remote Memory Access(RMA) semantics of the Message Passing Interface(MPI). We show that our framework is capable of achieving good solution quality with respect to the centralzied method while maintaining computational scalability.

In Section 2.2 we present current state of the art research pertaining to decentralized optimization approaches in power systems as well as the condition based monitoring approaches explored previously. We then discuss the decentralized sensor driven maintenance problem formulation in Section 2.3. In Section 2.4, we develop the decentralized sensor driven maintenance algorithm. In Section 2.5, we present our experimental results before presenting the conclusions and future work in 2.6.

## 2.2 Related work

All generators in a power network have a set of associated costs which influence its operational behavior over a planning horizon. Some of these costs include the *dispatch cost* which is the cost of production of a unit of energy per unit time, *commitment cost* which is the cost of turning on a generator or committing it to production, and *maintenance cost* which is the cost incurred in maintaining a generator by taking it out of service for a fixed period of time [8].

There are broadly two main categories of research within the domain of power systems: unit commitment [9] or operations, and maintenance [6]. Operations problem or the well known unit commitment problem determines which generators to turn on at which time periods subject to the commitment costs and determining how much electricity to produce based on the set of dispatch costs. Similarly, maintenance problem involves solving for an optimum time to perform maintenance on a generator.

Sensor based condition monitoring has been explored in [10] where the health of a device is estimated using degradation signals being captured by sensors. A Bayesian updating method is then used to predict the residual lifetime of the device. Condition monitoring is integrated in [3],[6] as a means to solve the maintenance problem mentioned above. Further, operations and maintenance are interlinked and modeled as an intelligent and condition aware optimization problem which when solved under certain operational constraints has been shown to yield a significant benefit to utility companies.

While the above mentioned works focus on solving a wide variety of problems in power systems, they are all centralized in their approach. None of them exploit the natural potential for parallelism which exists in power systems either in a spatial or a temporal manner. However, there have been numerous works which investigate the parallel aspect of problem solving in power systems to varying degrees.

Decentralized methods are an emerging sub category within distributed computing which are best suited for geographically distributed computing elements due to their fault-tolerant nature and high scalability [11]. Decentralized methods avoid a central co-ordinating node and are highly applicable to geographically scattered computing systems and their resulting applications.

Most decentralized optimization techniques study a subset of problems called the convex optimization problem. Decentralized, augmented Lagrangian techniques like ADMM [12] have been popular for quite some time, owing to their fast convergence properties [13] but their applicability has mostly been limited to solving convex problems. However, recent works have focussed on application of ADMM to non-convex and non smooth distributed optimization problems as well [14].

In the field of power systems, there have been quite a few attempts to seek a parallel solution of existing problems. The work done in [15] attempts to solve the stochastic unit commitment problem for a fleet of sustainable energy sources wherein the imposed demand is time varying. The paper formulates the unit commitment problem as a Mixed

Integer Programming(MIP) with the commitment variables being binary. It then decomposes the problem based on the generating energy resources and attempts to maximize the Lagrangian dual function in a parallel way to achieve minimum duality gap. The work done in [16] also uses a dual decomposition to solve a multi-area unit commitment problem. The authors in [17] also solve a unit commitment problem using an incremental sub-gradient method to progress the dual variable while simultaneously recovering the primal feasibility. Since the dual variables are smooth and differentiable the problem has been decomposed into a convex dual phase and a non convex primal phase and therefore a sub-gradient method proves beneficial.

A valid and practical issue in power system of large-scale unit commitment has been investigated in [18]. The paper focusses on solving the non-convex unit commitment problem subject to operational constraints by using dual decomposition using ADMM. The paper decomposes an existing problem into multiple operational regions or sub-problems each holding its own local optimization problem subject to local constraints. Each subproblem is assigned to one computing node and is solved in an iterative manner to achieve global convergence. Computing nodes are connected according to the adjacency graph spawned by the power network topology. The paper demonstrates convergence with varying degree of regions for the IEEE 3012 bus case.

In this chapter, we focus on the problem of generator maintenance and operations. We decompose our problem into various operational areas and achieve global convergence by solving the local optimization problem held by each. A novel and heretofore un-investigated aspect of our research stems from the sensor driven component of our optimization. By analyzing streaming sensor data from different generators across multiple regions, we generate an optimization model sensitive to the remaining lifetime for each generator. We solve this optimization model in a decentralized way and assign a near-optimum maintenance epoch to each asset while also providing a decision on production by each over a fixed planning horizon. Our approach provides a highly scalable method

that is amenable to dynamic changes to the generator health and is well suited for large scale power networks.

We now proceed towards formulating the optimization model and describing the network decomposition as well as the Lagrangian update scheme.

## 2.3 Problem Formulation

For obtaining the decentralized formulation, we decompose the problem based on regions leading to the respective regional subproblems. From a practical standpoint, each region may denote a subsidiary of the utility company in a vertically integrated market or a utility company in a deregulated market. Therefore, every region comprises of local generators and buses subject to its own operational constraint. Details about notations have been provided in Appendix A.

We demonstrate the regional decomposition scheme of a sample network 1 with the help of Figure 2.1. Our example network consists of 3 regions with boundary and foreign

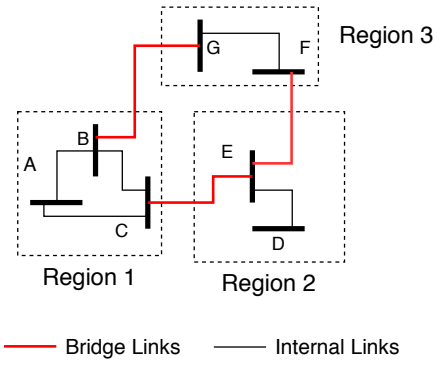



Figure 2.1: Partition of Network topology into regions.

bus categorization for each region defined as follows:

- Region 1: $\mathcal{U}_1 = \{B, C\}$, $\mathcal{V}_1 = \{G, E\}$
- Region 2: $\mathcal{U}_2 = \{E\}$, $\mathcal{V}_2 = \{F, C\}$
- Region 3: $\mathcal{U}_3 = \{G, F\}$, $\mathcal{V}_3 = \{B, E\}$

### 2.3.1 Objective Function

The regional subproblem seeks to minimize the objective cost as represented by Problem (2.1) subject to the constraints listed in Equations (2.4). For simplicity we consider the vector form of the variables as necessary.

$$\min_{\boldsymbol{\nu},\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\phi}} \quad \mathcal{L}_r(\bar{\boldsymbol{\theta}}_k, \bar{\boldsymbol{F}}_k, \boldsymbol{\lambda}_k, \boldsymbol{\phi}_k) = \sum_{t\in T}\sum_{g\in G_r} D_g y_t^g + M_g \nu_t^g + \sum_{t\in T}\sum_{b\in\mathcal{B}_r} [\lambda^{b,t}|\theta_t^b - \bar{\theta}_t^b| + \frac{\rho}{2}(\theta_t^b - \bar{\theta}_t^b)^2]$$
$$+ \sum_{t\in T}\sum_{u\in\mathcal{U}_r}\sum_{v\in\mathcal{V}_r^u} [\phi^{uv,t}|F_t^{uv} - \bar{F}_t^{uv}| + \frac{\rho}{2}(F_t^{uv} - \bar{F}_t^{uv})^2] \tag{2.1}$$

The objective function represented by Problem (2.1), comprises of an operations cost component, a maintenance cost component. In addition, it also comprises of ADMM penalty terms imposed on balancing flow estimates among neighboring regions. Flow estimates are iteratively balanced across transmission lines common with neighboring regions as well as with respect to the phase angles at their respective boundary buses.

### 2.3.2 Predictive Analytics

A crucial aspect of our approach is the ability to integrate the insights from sensor data into the regional operations and maintenance scheduling problem. In the objective function, the cost vector $M_g$ provides the critical link between sensor-driven predictive analytics and decision optimization by defining the expected cost of maintenance for generator $g$ as a function of the predictions on its remaining life distribution. The entries of $M_g$ are updated in real-time as new sensor data becomes available. The process involves a series of intermediate steps.

We first characterize the the degradation process in each generator $g$ using a parametric degradation function $D_g(t) = \phi_g(t;\kappa,\theta_g) + \epsilon_g(t;\sigma)$. In this formulation, $\phi_g(t;\kappa,\theta_g)$ defines the underlying base degradation function for generator $g$, and $\epsilon_g(t,\sigma)$ models the uncertainty due to degradation and measurement errors, with an associated variance pa-

rameter $\sigma$. $\kappa$ and $\theta_g$ denote the deterministic and stochastic degradation parameters, respectively. This parametric function is our sensor-based inference to the underlying degradation processes in the generators that ultimately lead to their failure. The failure of each generator is defined as the first time that their degradation signal reaches a predefined failure threshold $\Lambda_g$. The remaining life of generator $g$, namely $\tau_g$ can be evaluated as $P(\tau_g^f = t) = P(t = min(s \geq 0 | D_g(s) \geq \Lambda_g)$

We assume that engineering knowledge and historical data yields an initial estimate for the distribution of the stochastic degradation parameter $\theta_g$, denoted by $\pi(\theta_g)$. Further, we assume that the operators observe a set of degradation signals $\boldsymbol{d}_g^o$ at times $\boldsymbol{t}_g$. These observations enable us to update the distribution of the stochastic parameter $\theta_g$ to its posterior distribution counterpart $\upsilon(\theta_g)$ using Bayesian update:

$$\upsilon(\theta_g) = \left[ P(\theta_g | \boldsymbol{d}_g^o) = \prod_j P(D_g(t_j) = d_g^j | \theta_g, A_j) \right] \pi_i(\theta_g) / P(\boldsymbol{d}_g^o).$$

where $A_j$ denotes the condition that $D_g(t_k) = d_g^k$ for all $t_g^k \in \boldsymbol{t}_g^o$ such that $t_g^k < t_g^j$. Using these new definitions, we redefine the random variable for the remaining life $\tau_g$ as a function of the posterior distribution of the degradation parameter, $\upsilon(\theta_g)$, namely:

$$P(\tau_g > t) = \int P\left( \sup_{t_o \leq s \leq t_o + t} D_g(s) < \Lambda_g | \theta_g \right) \upsilon(\theta_g) d\theta_g. \tag{2.2}$$

where $t_o$ is the age of component.

Given the updated remaining life distributions, we derive the expected cost of conducting maintenance at time $t$, as follows:

$$C_{t^o,t}^g = \frac{c_g^p P(\tau_g > t) + c_g^f P(\tau_g \leq t)}{\int_0^t P(\tau_g > z) dz + t_o}, \tag{2.3}$$

where $c_g^p$ and $c_g^f$ are the costs for preventive maintenance, and unexpected failure for generator $g$, respectively. The function translates the remaining life distribution of generator

$g$ into a degradation-based function of cost over time. The cost vectors $M_g \ \forall g \in \mathcal{G}$ are discretized forms of this function. Please see [3, 6] for a more detailed explanation of the predictive analytics framework.

### 2.3.3 Constraints

The regional subproblem for the joint operations and maintenance optimization is subject to a number of local constraints:

$$\sum_{t \in T} \nu_t^g = 1 \quad \forall g \in \mathcal{G} \tag{2.4a}$$

$$[1 - \nu_t^g] C_g \geq y_t^g \quad \forall g \in \mathcal{G}, \forall t \in T \tag{2.4b}$$

$$\Gamma^{uv}(\theta_t^u - \theta_t^v) = F_t^{uv} \quad \forall u \in \mathcal{U}_r, \forall v \in \mathcal{V}_r^u, \forall t \in T \tag{2.4c}$$

$$f_{min}^{uv} \leq F_t^{uv} \leq f_{max}^{uv} \quad \forall u \in \mathcal{U}_r, \forall v \in \mathcal{V}_r^u, \forall t \in T \tag{2.4d}$$

$$\sum_{g \in G_{r,b}} y_t^g - d_t^b = \sum_{b' \in B_b^r} [\Gamma^{uv}(\theta_t^b - \theta_t^{b'})] \quad b \in B^r, b' \in B_b^r \tag{2.4e}$$

Constraint (2.4a) imposes the constraint that maintenance must be performed on each generator exactly once during the planning horizon. Constraint (2.4b) ensures that a generator that has been placed under maintenance must not have any production. Constraints (2.4c), (2.4d) calculate flow values based on phase angles and ensure adherence to transmission line capacities. Constraint (2.4e) attains network flow balance with respect to local demand and production at every bus within the region.

### 2.3.4 Lagrangian Update

We estimate two important quantities, i.e. *intermediate flow* denoted by $\bar{F}_t^{uv}$ and *intermediate phase angles* denoted by $\bar{\theta}_t^b$ which are calculated based on the phase angle estimates received from neighboring regions $\forall b \in \mathcal{B}_r$ and $\forall u \in \mathcal{U}_r, \forall v \in \mathcal{V}_r$ for phase angles and

flows respectively applied $\forall t \in T$.

$$\bar{\theta}_t^b = \frac{\theta_t^b + \sum\limits_{r' \in \mathcal{N}_r^b} \tilde{\theta}_t^{b,r'}}{|\mathcal{N}_r^b|+1} \quad b \in B^r \tag{2.5}$$

$$\bar{F}_t^{uv} = \frac{\Gamma^{uv}(\tilde{\theta}_t^{u,r'} - \tilde{\theta}_t^{v,r'}) - \Gamma^{uv}(\theta_t^u - \theta_t^v)}{2}, \quad u \in \mathcal{U}_r, v \in \mathcal{V}_r^u, r \in \mathcal{N}_r^u \tag{2.6}$$

We update the Lagrangian multipliers as follows

$$\lambda^{b,t} = \lambda^{b,t} + \rho(\theta_t^b - \bar{\theta}_t^b) \quad \forall b \in \mathcal{B}_r, \forall t \in T \tag{2.7}$$

$$\phi^{uv,t} = \phi^{uv,t} + \rho(F_t^{uv} - \bar{F}_t^{uv}) \quad \forall u \in \mathcal{U}_r, \forall v \in \mathcal{V}_r, \forall t \in T \tag{2.8}$$

The optimization model given by Problem (2.4) describes a Mixed-Integer Quadratic Problem (MIQP) which solves for the maintenance and operations in a decentralized manner. Since we have the presence of binary variables in $\nu$, our problem is *non-convex*. As a result, it becomes much harder than traditional convex schemes to achieve convergence in a decentralized manner. However, recent works as mentioned in the previous sections have demonstrated the successful application of ADMM for solving decentralized non-convex problems. The maintenance cost is fed as input to the data and is derived from the work done in [6]. We now proceed to describing the decentralized algorithm and its implementation.

The Lagrangian terms in the model serve as *penalties* for deviating from a position of balance. Convergence occurs when these terms tend to become small enough such that the optimization problem given by Problem (2.4) become mathematically equivalent to that of a *centralized* problem as described in [6]

## 2.4 Decentralized Maintenance and Operations Algorithm

**Algorithm 1** Decentralized Maintenance and Operations Algorithm

Initialize $\boldsymbol{\nu}_0^g, \boldsymbol{y}_0^g, \forall g \in G_r$
Initialize $\boldsymbol{\theta}_0^b, \tilde{\boldsymbol{\theta}}_0^b, \bar{\boldsymbol{\theta}}_0^b, \forall b \in \mathcal{B}_r$
Initialize $\boldsymbol{F}_0^{uv}, \tilde{\boldsymbol{F}}_0^{uv}, \bar{\boldsymbol{F}}_0^{uv}, \forall u \in \mathcal{U}_r v \in \mathcal{V}_r$
$GC \leftarrow 0$ // global convergence state
$LC_r \leftarrow 0$ // local convergence state of region $r$
**while** $\neg GC$ **do**
  $\{\boldsymbol{\nu}_{k+1}, \boldsymbol{y}_{k+1}, \boldsymbol{\theta}_{k+1}, \boldsymbol{F}_{k+1}\} \leftarrow \mathcal{L}_r(\bar{\boldsymbol{\theta}}_k, \bar{\boldsymbol{F}}_k, \boldsymbol{\lambda}_k, \boldsymbol{\phi}_k)$
  Generate $\lambda_{k+1}^{b,t}, \phi_{k+1}^{uv,t}$ based on (2.7) and (2.8) respectively
  Send $\boldsymbol{\theta}_k^b$ to all regions $r' \quad \forall r' \in \mathcal{N}_r, b \in \mathcal{B}_{r'}$
  Recieve $\tilde{\boldsymbol{\theta}}_k^b$ from all regions $r' \quad \forall r' \in \mathcal{N}_r, b \in \mathcal{B}_{r'}$
  Compute $\bar{\boldsymbol{\theta}}^b, \quad \forall b \in \mathcal{B}_r$ using (2.5)
  Compute $\bar{\boldsymbol{F}}_k^{uv}, \quad \forall u \in \mathcal{U}_r v \in \mathcal{V}_r$ using (2.6)
  **if** $||\boldsymbol{F}_{k+1} - \bar{\boldsymbol{F}}_k|| < \epsilon_p$ and $||\bar{\boldsymbol{F}}_{k-1} - \bar{\boldsymbol{F}}_{k-1}|| < \epsilon_d$ **then**
    $LC_r \leftarrow 0$
    **if** $\sum_{\forall r \in \mathcal{R}} LC_r = |\mathcal{R}|$ **then**
      $GC \leftarrow 1$
    **end if**
  **end if**
**end while**

The algorithm for performing decentralized optimization given in Problem (2.4) is given by 1. The algorithm describes the steps taken by a single region to solve the optimization problem held locally. In 1, every region initializes its local variables including the global and local convergence states. At the $k^{th}$ iteration, the regional Problem (2.1) is solved subject to constraints (2.4a)-(2.4e). The local phase angle and flow estimates are exchanged with neighboring regions and the estimates of the intermediate flow and phase angle values are updated according to Equations (2.5), (2.6). The region is deemed to have locally converged if the primal and dual variables with respect to the flow have fallen below certain thresholds denoted by $\epsilon_p, \epsilon_d$ respectively. The regions perform a global reduction, that sums over their individual local variable states in order to obtain the global convergence state of the entire system. If every region has attained local convergence, the algorithm terminates with the optimal solution.

We now proceed to describe our implementation and the results we have acquired over various test cases.

## 2.5 Results

### 2.5.1 Experimental Objectives

In order to implement 1 we needed a robust and versatile testing environment that could easily handle multiple testing scenarios and be scalable as well. Further, we needed a good set of communication libraries which would work in a distributed computing environment reliably with minimal effort. An indigenous TCP/IP based communication paradigm would have required a significant amount of development and debugging effort and would have been difficult to scale in a decentralized manner. Further, it would not have provided a rich set of communication primitives that would be required to solve our optimization framework in a decentralized manner.

Therefore, we used MPI for simulating our algorithm, wherein each region in our problem was assigned to a process. Using a distributed memory model like MPI for communication helps us evaluate the algorithm on a near *real-world* setting, where the regions may represent individual participants of an RTO as a collective. Further, we impose an overlay network on top of the MPI layer which restricts a particular node to communicate only with the nodes representing the neighbors for its own region.

The software framework developed as part of this project is in itself not meant to be a holistic software package to solve decentralized maintenance problems for power networks. Rather, it is a means to validate our algorithm for different test cases and parameters with the promise of high scalability on an experimental basis. As we will point out in later sections, we also intend to extend this software framework to a multi-cluster setup connecting clusters using TCP/IP stack for further validation of the algorithm under more realistic settings.

Table 2.1: IEEE 118 centralized solution

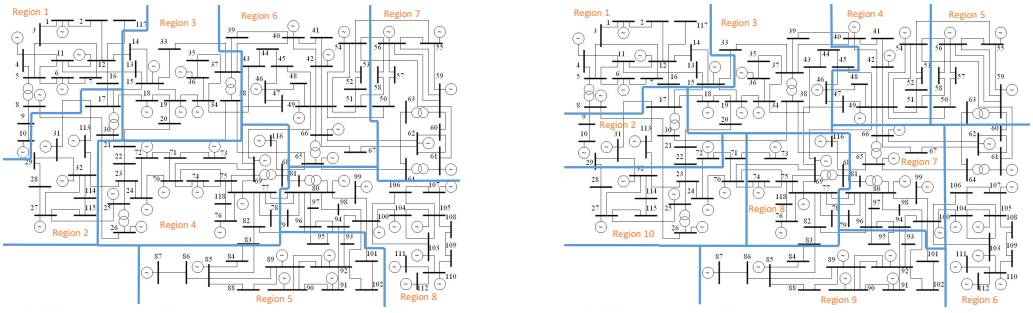
| CL | $\rho$ | $\zeta_{oper}$ | $\zeta_{maint}$ | $\zeta_{total}$ | $\Delta$ | Time(in secs) |
|---|---|---|---|---|---|---|
| - | - | 6048265 | 1606428 | 7654693 | - | 5.4 |

Table 2.2: IEEE 118 bus 8 region case: decentralized results

| CL | $\rho$ | $\xi_{oper}$ | $\xi_{maint}$ | $\xi_{total}$ | $\Delta$ | Time(in secs) |
|---|---|---|---|---|---|---|
| 5 | 0.5 | 6103568 | 1606428 | 7709997 | 0.7% | 309 |
| 5 | 0.2 | 6067177 | 1606428 | 7673605 | 0.2% | 670 |
| 10 | 0.2 | 6073533 | 1606428 | 7679961 | 0.3% | 548 |
| 5 | 0.1 | 6049028 | 1606428 | 7655457 | 0.01% | 1055 |
| 10 | 0.1 | 6054384 | 1606428 | 7660813 | 0.07% | 664 |

### 2.5.2 Experimental Setup

We used `python` as the programming language for the framework. The `mpi4py` [19] package which is an MPI package for `python` was used to build the decentralized framework. We used `Gurobi 6.5` [20] for solving the MIQ problem represented by Problem (2.4) on each node. We evaluate our model on the IEEE 118 bus case with data derived from the `pypowerAPI` [21]for python which is the `python` analogue of `MATPOWER` library of `MATLAB` [22] .




(a) 8 regions (b) 10 regions

Figure 2.2: Decomposition of IEEE 118 case with 8 regions and 10 regions

In order to benchmark our result, we execute a centralized version of Algorithm 1 us-

Table 2.3: IEEE 118 bus 10 region case: decentralized results

| CL | $\rho$ | $\xi_{oper}$ | $\xi_{maint}$ | $\xi_{total}$ | $\Delta$ | Time(in secs) |
|---|---|---|---|---|---|---|
| 5 | 1 | 6104844 | 1606428 | 7711272 | 0.7% | 464 |
| 10 | 0.5 | 6080938 | 1606428 | 7687336 | 0.4% | 473 |
| 10 | 0.075 | 6050101 | 1606428 | 7656529 | 0.02% | 2242 |

Table 2.4: IEEE 3012 centralized solution

| CL | $\rho$ | $\zeta_{oper}$ | $\zeta_{maint}$ | $\zeta_{total}$ | $\Delta$ | Time(in secs) |
|---|---|---|---|---|---|---|
| - | - | 32379937 | 15226705 | 47606643 | - | 582 |

ing only 1 region comprising all the assets and rate the performance of the decentralized algorithm using the optimality gap $\Delta$

$$\Delta = |1 - \frac{\xi_{total}}{\zeta_{total}}| \tag{2.9}$$

where

$$\xi_{total} = \xi_{maint} + \xi_{oper} \tag{2.10}$$

$$\zeta_{total} = \zeta_{maint} + \zeta_{oper} \tag{2.11}$$

where $\xi_{total}, \xi_{maint}, \xi_{oper}$ represent the total objective function value, its operations component and its maintenance component measured on convergence for the decentralized case. Similarly $\zeta_{total}, \zeta_{maint}, \zeta_{oper}$ represents total objective function value, its operations component and its maintenance component for the centralized case. We set the maintenance cost coefficient(MCC) proportional to the dispatch cost leading to the relation

$$\mu_g = KD_g, \forall g \in \mathcal{G} \tag{2.12}$$

Table 2.5: IEEE 3012 60 regions decentralized solution

| CL | $\rho$ | $\zeta_{oper}$ | $\zeta_{maint}$ | $\zeta_{total}$ | $\Delta$ | Time(in secs) |
|---|---|---|---|---|---|---|
| 10 | 5 | 33311976 | 15226705 | 48538682 | 1.9% | 37525 |

where K is a constant set to the value 1000. We also make our power line capacities unbounded such that $F_t^{uv} \in (-\infty, \infty)$. We set the dispatch cost of the generators such that

$$D^g \in [20, 40] \tag{2.13}$$

The above set of costs were chosen based on their ability to yield a healthy ratio between operations and maintenance as determined empirically. We define $\alpha$ and $\beta$ as a means to define the primal and dual residuals.

$$\alpha = \sum_{t \in T} \sum_{u \in \mathcal{U}_r} \sum_{v \in \mathcal{V}_r^u} |F_{t,k}^{uv} - \bar{F}_{t,k}^{uv}|^2 \tag{2.14}$$

$$\beta = \sum_{t \in T} \sum_{u \in \mathcal{U}_r} \sum_{v \in \mathcal{V}_r^u} |F_{t,k}^{uv} - F_{t,k-1}^{uv}|^2 \tag{2.15}$$

For termination we impose the criteria $\alpha < \epsilon_p$ and $\beta < \epsilon_d$ where $\epsilon_p$ and $\epsilon_d$ are the tolerance on the primal and dual residuals. From the data presented in Table 2.2 and Table 2.3, we can make a number of observations. Firstly, the decentralized algorithm takes much longer than its centralized counterpart. This is expected for the IEEE 118 bus case. However, initial results obtained from scaling to a much bigger IEEE 3012 bus case shows that the centralized algorithm will fail to yield a solution within a practical duration of time. Secondly, the problem is extremely sensitive to the penalty parameters. In fact, the same argument can be extended towards the costs present in the objective function as a whole itself. It can be shown empirically that the problem will fail to converge if the penalty parameters aren't chosen properly. Lastly, by setting the $\rho$ to a lesser value provides a higher

accuracy, at the cost of more time. As expected a higher convergence limit results in a faster convergence. However, we note that a higher convergence limit has little effect on accuracy in terms of the optimality gap. It is also worth noting that above results have been obtained or the predetermined set of costs specified above, a deviation from above costs would require re-calibration in terms of the penalty parameter $\rho$.

## 2.6 Conclusion

In this chapter we develop a decentralized algorithm for performing generator maintenance and dispatch. We decompose a given power network topology into multiple regions which may represent different power companies and use the idea of Lagrangian multipliers to formulate an optimization model. We solve the optimization model in a decentralized manner wherein each region holds its own local optimization model and colludes with its neighbors to achieve convergence. We implement this algorithm using MPI wherein we assign each region to an MPI node and use an overlay network which determines the neighboring nodes based on their corresponding regions. We further demonstrate the convergence of the algorithm on the IEEE 118 bus case with 8 regions and 10 regions. Our results indicate a very high sensitivity to the penalty parameter as well as the costs.

A logical next step is to explore the asynchronous version of Algorithm 1. There are a number of challenges in this regard and significant progress has been made in order to solve them. Firstly, asynchronous communication is much harder due to an imbalance in the frequency of communication across various regions. A conventional Send/Receive of MPI fails to work in this regard since some regions send more messages than others resulting in a buffer congestion on the remote node thereby delaying the progress of the algorithm. To address this we have incorporated the Remote Memory Access (RMA) scheme offered by MPI. This further brings up the chance of contention among regions to acquire value stored on a remote node leading to delays in the algorithm. Another important aspect for an asynchronous algorithm is a fully decentralized termination mechanism. There needs to

be a way for each node in the network to have knowledge of the status of convergence of all nodes in the network in a decentralized and asynchronous manner. These above highlighted issues serve as key to realizing a robust asynchronous decentralized optimization algorithm.

Our algorithm shows good potential for increasing power network efficiency by solving large scale decentralized optimization problems successfully.

# CHAPTER 3
# ASYNCHRONOUS DECENTRALIZED FRAMEWORK FOR UNIT COMMITMENT IN POWER SYSTEMS

## 3.1 Introduction

The Unit Commitment (UC) problem [9] is one of the key optimization problems that has received considerable attention over the past few decades. The goal of the UC problem is to determine how much power each generator has to produce given the cost of production and the power demand to be met.

The UC problem is a critical component for daily production planning for power utility companies. Utility companies that command power networks typically solve the UC problem many times throughout the entire day to determine the production levels of different generators on an hourly basis. The power network is subject to demand uncertainties that require constant monitoring and adjustments to power generation. There may also be sudden and unpredictable outages caused by natural or man-made phenomena that may disrupt the highly sensitive, hourly production schedule of the generators network-wide. An efficient algorithm for solving the UC problem therefore needs to be highly robust to sudden changes in operating conditions. The UC problem involves binary decision variables to decide which generators must be turned on or off at various time epochs across a fixed planning horizon. The resulting optimization model of the UC problem therefore becomes a Mixed Integer Programming (MIP) problem that falls under the category of non-convex optimization problems.

Existing approaches to decentralized unit commitment problems [18] adopt a synchronous approach wherein each iteration of the local subproblem and the subsequent information exchange is performed in tandem by all regions. Owing to a tight coupling between

computation and communication, the information obtained from neighboring regions in the synchronous approach is always pertaining to the current iteration. Such synchronous models do not account for geographically distant computational nodes wherein transfer of data between nodes comes with its own communication delay. At different nodes, different subproblems and computing architectures with varying computational capabilities will lead to heterogenous processing times. This variability in the computation time would be further compounded by time-varying loading on the computing resources due to on-line control of many connected assets in the region. Therefore, methods that rely on synchronization of computational nodes do not account for delays incurred in practice. Consequently, in large-scale distributed systems involving many interconnections, achieving perfect synchrony is extremely difficult [23]. Furthermore, synchronous computational methods may simply fail to converge in the face of even a slight degree of asynchrony [24]. Even with a hypothetical synchronous computation system, a slow computational node or communication link can significantly increase the computational time by under-utilizing the computing resources. This causes all the computational nodes to wait until the problematic node completes the computation and data transfer. In addition, online instrumentation and digital control might make power systems vulnerable to cyber attacks and with the assumption of synchronization, any attack to stop or slow down the operations of a single local node can significantly impact global power system operations.

Asynchronous methods can be used to circumvent the aforementioned problems as well as to develop a more flexible and resilient computational platform. We stand to gain significant computational benefit by adopting an asynchronous approach wherein those regions with a smaller subproblem to solve may proceed onto their next iteration without waiting on the slower regions to finish. In doing so, each region executes its iterations independently of other regions and uses the latest available information from its neighbors for each of its iterations. In an asynchronous approach, the solution of the entire system is no longer held up by a slow computational node and a faster solution to the global UC problem can

be expected.

Non-convex, MIP problems are challenging to solve in their own right. However, asynchronous nature adds another layer of complexity to such problems and their convergence behavior is still an open question in the optimization community. To the best of our knowledge, this work is the first to attempt to outline an asynchronous decentralized MIP based framework to solve the UC problem. With a region based decomposition, our technique asynchronously solves for an optimal production schedule within each region while ensuring privacy as well as minimal sharing of commercial data between any two regions. We iteratively use the Alternating Direction Method of Multipliers (ADMM) to solve the UC problem locally and exchange information with neighbors, eventually converging to a solution close to that of the centralized problem. Our method can be orchestrated on cloud based infrastructure, institutional clusters or a distributed hybrid combination of both. We show that our method is capable of handling unbalanced computation among regions and computationally performs better than its synchronous counterpart without compromising on solution quality to a great extent.

This chapter is organized as follows. In Section 3.2 we discuss approaches in power systems that are pertinent to the UC problem. We then proceed towards the centralized and decentralized mathematical formulations of the UC problem in Section 3.3. We describe our novel asynchronous decentralized algorithm for solving the UC problem in Section 3.4. We detail the experimental setup used to validate our algorithm as well as discuss the results of our algorithm on the IEEE 118 bus case in Section 3.5. We present our concluding remarks in Section 3.6.

## 3.2 Related Work

In the field of power systems, owing to the complexity of MIP UC problems, there have been several attempts to seek a parallel solution of the UC problem. The work done in [15] attempts to solve the stochastic unit commitment problem for a fleet of sustainable energy

sources wherein the imposed demand is time varying. The work done in [16] also uses a dual decomposition to solve a multi-area unit commitment problem in a synchronous fashion. The work done in [17] solves a unit commitment problem using a two stage approach via an incremental sub-gradient method to progress the dual variable while simultaneously recovering the primal feasibility.

Decentralized methods form a fast emerging sub category within distributed computing which are best suited for geographically distributed computing elements due to their fault-tolerant nature and high scalability [11]. In the context of decentralized approaches, augmented Lagrangian techniques like ADMM [12] have been popular for quite some time, owing to their fast convergence properties [13], but their applicability has mostly been limited to solving convex problems. However, recent works have focussed on application of ADMM to a fixed set of non-convex and non smooth distributed optimization problems under some strict conditions [14]. The work done in paper [18] investigates large-scale decentralized unit commitment by solving the non-convex unit commitment problem subject to operational constraints by using dual decomposition and ADMM in a synchronous manner. The work done in the paper [25] demonstrates convergence of asynchronous ADMM in a distributed master-slave based computing model.

Our algorithm differs significantly from all approaches presented in this section. We primarily employ a region based topological decomposition of the power network and solve each region's sub problem in a decentralized and asynchronous fashion. Our problem is not a consensus optimization problem and owing to its asynchronous and non-convex nature is much more challenging to solve. In this chapter we demonstrate that our algorithm provides a good solution quality in a decentralized and asynchronous setting. We now proceed to the problem formulation.

### 3.3 Problem Formulation

The UC problem is usually solved over a fixed planning horizon of size $T$, divided into discrete time epochs. We denote the set of all generators as $\mathcal{G}$ and the set of buses as $\mathcal{B}$. The UC optimization problem can be represented as follows:

$$\min_{\boldsymbol{x},\boldsymbol{y}} \quad \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{d}^\top \boldsymbol{y} \tag{3.1a}$$

$$\text{subject to} \quad \boldsymbol{A}\boldsymbol{y} + \boldsymbol{B}\boldsymbol{x} = \boldsymbol{E} \tag{3.1b}$$

$$\boldsymbol{F}\boldsymbol{y} = \boldsymbol{H} \tag{3.1c}$$

where $\boldsymbol{x}$ is a vector of length $|\mathcal{G}|\times T$ whose components are binary values specifying if generators are turned on or turned off across each time epoch for each generator in the network. Similarly, $\boldsymbol{y}$ is a real-valued vector of length $(|\mathcal{B}|+|\mathcal{G}|) \times T$ whose components represent dispatch variables specifying the level of production on generators as well as the electric phase angles on separate buses.

The objective function (3.1a) in Problem (3.1) involves two terms $\boldsymbol{c}^\top \boldsymbol{x}$ and $\boldsymbol{d}^\top \boldsymbol{y}$ which represent the cost associated with turning generators on or off across the planning horizon, referred to as commitment, and the cost incurred in production of electricity respectively. The objective is to minimize this total cost across the entire network subject to constraints that either focus on every generator separately as in (3.1b), or link all the generator operations together as in (3.1c). The first class of constraints given by (3.1b) include limits on generator production capacities and embody flexibilities such as how fast production can change within an hour. The second set of constraints given by (3.1c) ensure that network demand is satisfied while ensuring that electricity flow is in line with transmission capacities. We refer the interested reader to [6] [18] for a more detailed formulation of the UC problem.

Given a region based partition, we can divide the UC problem stated in Problem (3.1) into smaller region based subproblems. In such a case the objective function of each subproblem only involves variables dealing with assets of a particular region. Similarly, each region also ends up holding its own set of constraints corresponding to (3.1b), (3.1c) involving assets specific to that region alone. Therefore, given a region based partition, the UC problem may be solved in an entirely decentralized manner where each processing element handles the UC sub-problem of a particular region. Every region iteratively tries to obtain a balance in terms of the dispatch variables with respect to the subproblem residing in its neighboring regions.

We define $\mathcal{G}_r$ as the set of generators in region $r$ and $\mathcal{R}_r$ as the set of neighboring regions of $r$. Similarly, $\boldsymbol{x}_r, \boldsymbol{y}_r$ represent the binary commitment and dispatch variables of region $r$. In the decentralized case, ADMM is applied by relaxing constraint (3.1c). ADMM iteratively uses Lagrangian penalty terms $\boldsymbol{\lambda}$ and $\rho$ to ensure that the final solution satisfies this constraint within acceptable accuracy. Therefore, ADMM ensures that flow of electricity is balanced between regions causing production of each region to adjust accordingly thereby satisfying demand per region indirectly for the entire network.

For each region $r$, we obtain $\hat{\boldsymbol{y}}_{r'}$ which is an estimate of $\boldsymbol{y}_{r'} \quad \forall r' \in \mathcal{R}_r$, and solve the following problem:

$$\min_{\boldsymbol{x}_r, \boldsymbol{y}_r} \quad \boldsymbol{c}_r^\top \boldsymbol{x}_r + \boldsymbol{d}_r^\top \boldsymbol{y}_r + \boldsymbol{\lambda}_r^\top (\boldsymbol{F}_r \boldsymbol{y}_r + \sum_{r' \in \mathcal{R}_r} \boldsymbol{F}_{r'} \hat{\boldsymbol{y}}_{r'} - \boldsymbol{H}_r) + \rho \left|\left| \boldsymbol{F}_r \boldsymbol{y}_r + \sum_{r' \in \mathcal{R}_r} \boldsymbol{F}_{r'} \hat{\boldsymbol{y}}_{r'} - \boldsymbol{H}_r \right|\right|_2^2 \tag{3.2a}$$

$$\text{subject to} \quad \boldsymbol{A}_r \boldsymbol{y}_r + \boldsymbol{B}_r \boldsymbol{x}_r = \boldsymbol{E}_r \tag{3.2b}$$

$$\tag{3.2c}$$

where $\boldsymbol{\lambda}_r$ is updated as follows.

$$\boldsymbol{\lambda}_r = \boldsymbol{\lambda}_r + \rho(\boldsymbol{F}_r \boldsymbol{y}_r + \sum_{r' \in \mathcal{R}_r} \boldsymbol{F}_{r'} \hat{\boldsymbol{y}}_{r'} - \boldsymbol{H}_r) \tag{3.3}$$

With the decentralized formulation of the UC problem presented above, we now move on to describe the asynchronous decentralized UC algorithm.

### 3.4 Algorithm Design

As stated in Section 3.1, solving the non-convex decentralized UC problem is not an easy task. However, it is much easier to solve a convex version of the same problem [18]. The optimization model given in Problem(3.2) can be relaxed into a convex case by removing the binary constraint for commitment variables in $\boldsymbol{x}_r$ and making them continuous in the interval $[0, 1]$. We can use the solution from the relaxed version as a good initial guess to solve the actual decentralized UC optimization model given by Problem (3.2) with the binary condition enforced. Therefore we solve the decentralized UC optimization model given by Problem (3.2) in two phases. It is important to note that both phases are solved in a purely asynchronous and decentralized manner.

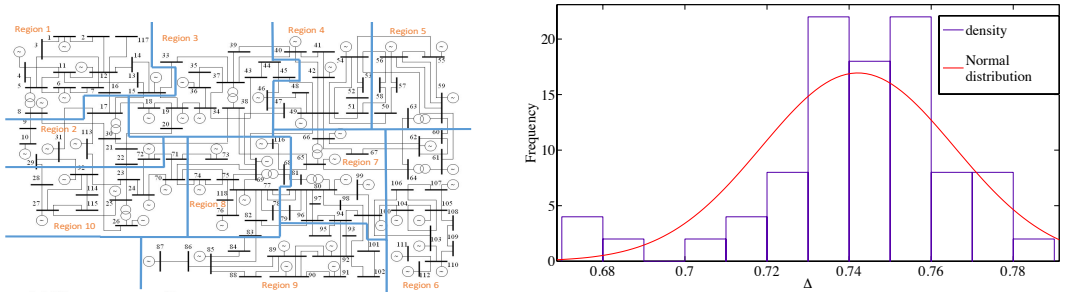
Algorithm 2 represents the asynchronous decentralized algorithm for solving the UC problem which is solved in convex and non-convex phases. In the former phase, we relax the binary commitment vector $\boldsymbol{x}$ such that all its elements lie in the continous interval $[0, 1]$ making Problem (3.2) convex whereas in the latter phase, we impose the constraint such that $\boldsymbol{x} \in \{0, 1\}$ thereby solving its non-convex version. Each node then receives the dispatch estimate $\hat{\boldsymbol{y}}_{r'}$ from each of its neighboring region $r'$. Local convergence in either case is determined if the norm of the primal and dual dispatch variables falls below a fixed threshold denoted by $\alpha$ and $\beta$ respectively. Global convergence occurs when every region attains local convergence. We designate a master node to determine global convergence using local convergence values of worker nodes asynchronously. The algorithm terminates

**Algorithm 2** Asynchronous Decentralized UC Algorithm run on each processor

Initialize $\boldsymbol{x}_r, \boldsymbol{y}_r$
$k \leftarrow 0$
**for** phase = convex,non-convex **do**
  **while** global convergence is not acheived **do**
    send $\boldsymbol{y}_r$ to regions $\forall r' \in \mathcal{R}_r$
    in convex phase, set $\boldsymbol{x}$ such that $0 \leq x_i \leq 1, \forall x_i \in \boldsymbol{x}$
    in non-convex phase, set $\boldsymbol{x}$ such that $x_i \in \{0, 1\}, \forall x_i \in \boldsymbol{x}$
    solve Problem (3.2)
    compute $\bar{\boldsymbol{y}}_r^k = \sum_{r' \in \mathcal{R}_r} \boldsymbol{F}_{r'} \hat{\boldsymbol{y}}_{r'}$
    update $\boldsymbol{\lambda}$ according to (3.3)
    **if** $(||\boldsymbol{y}_r - \bar{\boldsymbol{y}}_r^k|| < \alpha)$ and $(||\bar{\boldsymbol{y}}_r^k - \bar{\boldsymbol{y}}_r^{k-1}|| < \beta)$ **then**
      local_convergence $\leftarrow 1$
    **else**
      local_convergence $\leftarrow 0$
    **end if**
    $k \leftarrow k + 1$
  **end while**
**end for**

upon attaining global convergence.

### 3.5 Experimental Setup and Results




(a) 10 region partition of IEEE 118 bus case (b) Distribution of $\Delta$, with $\mu = 0.74, \sigma = 0.0235$

Figure 3.1: IEEE 118 bus case[1]

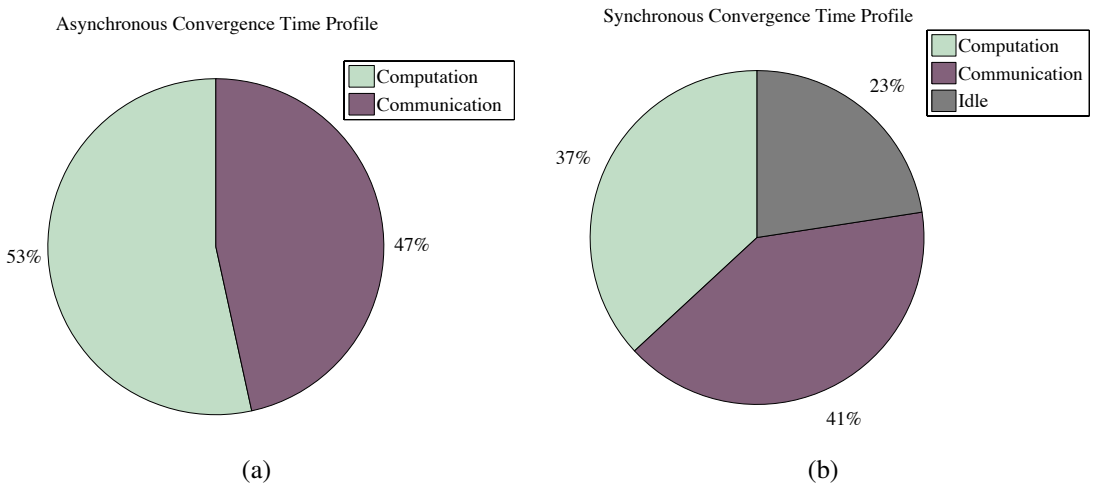
We simulate a large power network topologically divided into 10 regions based on the IEEE 118 bus case involving 54 generators derived from *MATPOWER* [22]. We solve the unit commitment problem over a planning horizon comprising of 24 time epochs spanning

[1] http://energy.komisc.ru/dev/test˙cases

an entire day. The objective behind this simulation is to determine convergence behavior of Algorithm 2 owing to its non-convex nature. We simulate a geographically dispersed set of nodes on a high performance cluster comprising of Intel Xeon CPUs with a clock rate of 2.80GHz with each core representing one region. We used Remote Memory Access (RMA) primitives of MPI with passive target synchronization to facilitate asynchronous message passing. The *mpi4py* [19] package was used as an interface to the MPI library for our experiments. *Gurobi 6.5* [20] was used for performing the optimization with respect to Problem (3.2) on each node. We evaluate Algorithm 2 with respect to its synchronous counterpart obtained by inserting an MPI *Barrier* call at the beginning of each iteration of Algorithm 2.

An asynchronous algorithm is stochastic in nature, therefore we performed a total of 50 runs of Algorithm 2 with the same run time parameters and recorded the communication and the computation time for each of the runs with $\alpha = 5$ and $\beta = 10$. For the synchronous version, apart from the communication and computation time, we also record the idle time each processor spends owing to the *Barrier* call. In order to establish a high degree of asynchrony, we add a small delay of 0.2 secs for half of the regions in both cases. We measure the degree of asynchrony, $\Delta$, on the basis of the ratio between the least number of iterations vs the maximum number of iterations performed by any node in the asynchronous case. Figure 3.1(b) shows the distribution of $\Delta$ measured over all the 50 runs. A mean value of 0.74 implies that for every 100 iterations performed by the fastest region in the system, the slowest region performs on an average close to 74 iterations. Therefore, Figure 3.1(b) demonstrates that the 10 region partition of the 118 bus case has a high degree of asynchrony.

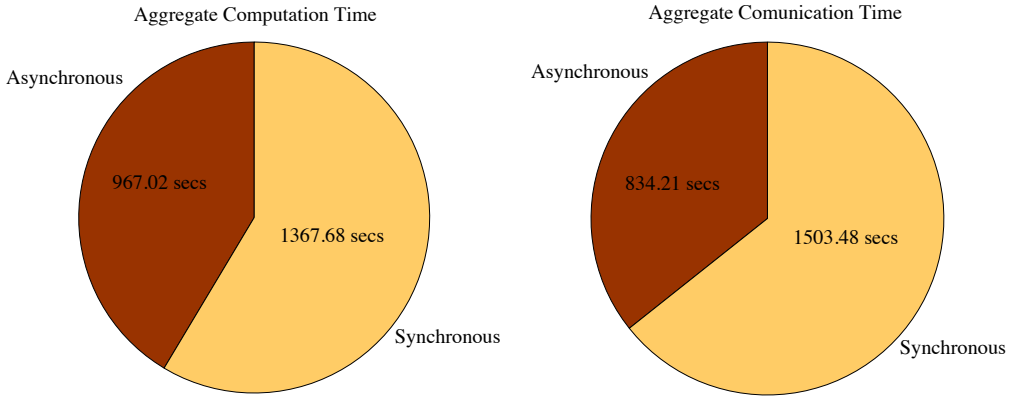
Figure 3.2(a) pictorially depicts data pertaining to the fraction of time spent by the processors performing communication and computation tasks in the asynchronous case. Figure 3.2(b) shows the fraction of time spent idle by the processors in addition to the fraction of time spent in communication and computation tasks in the synchronous case.




(a) (b)

Figure 3.2: Analysis of Convergence Time profile for Asynchronous and Synchronous Methods

From Figures 3.2(a), 3.2(b) we can infer that the asynchronous version of Algorithm 2 spends a substantial part of its time indulging in actual computation tasks as compared to its synchronous counterpart. Moreover, it can also be noted from Figure 3.2(b) that in the synchronous algorithm nodes spend a significant amount of time in an idle state.




(a) (b)

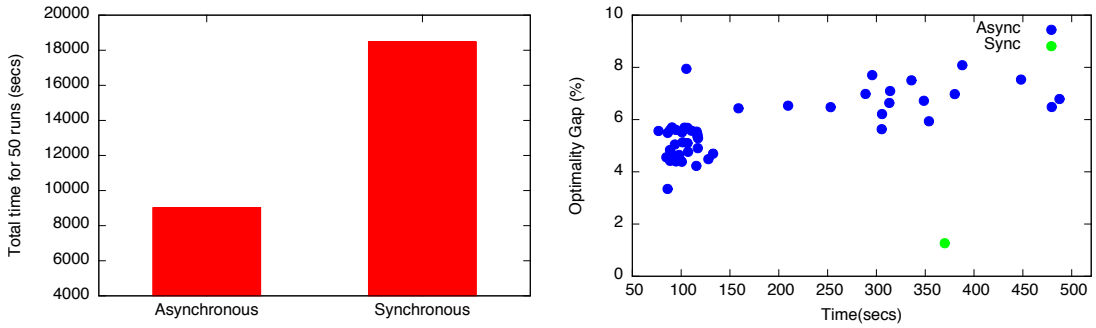
Figure 3.3: Analysis of aggregate total CPU time spent in Asynchronous and Synchronous methods

We calculate the average communication and computation time for each of the nodes and obtain an aggregate of total CPU time spent in each task by summing over the respective average values for all the nodes. We present the aggregate values of CPU time sepa-

Table 3.1: Convergence time (in secs) analysis of Algorithm 2

| Synchronous | | Asynchronous | |
|---|---|---|---|
| Mean ($\mu$) | Std. Dev. ($\sigma$) | Mean ($\mu$) | Std. Dev. ($\sigma$) |
| 369.57 | 0.72 | 180.40 | 122.02 |

rately for the computation as well as communication related tasks for both synchronous and asynchronous versions of Algorithm 2 in Figure 3.3. From Figure 3.3(a) and Figure 3.3(b) we can see that the total aggregate CPU time spent in both computation and communication is in fact lesser in the asynchronous case than that of the synchronous version. Therefore, from Figure 3.3 we can infer that Algorithm 2 promises faster convergence while tolerating a good degree of asynchrony while a synchronous version incurs a significant amount of idle time. From Figures 3.2, 3.3 we also infer that even though the asynchronous method spends lesser time performing computation related tasks, it is computationally more efficient than the synchronous case as the fraction of time spent in computation is higher for the asynchronous case.




(a) Cumulative Behavior (b) Optimality Gap

Figure 3.4: Performance Analysis of Algorithm 2

Table 3.1 shows the mean $\mu$ and standard deviation ($\sigma$) of 50 runs of both the synchronous and the asynchronous versions of Algorithm 2 with the same parameters. From Table 3.1 we can see that the synchronous case exhibits very little variation in convergence time and is relatively consistent across all runs whereas the asynchronous case shows considerable variation in run time. However, we can also infer that on an average, the

asynchronous method is almost twice as fast as the synchronous method and convergence time values within $1.5\sigma$ of the mean in the asynchronous case are still lower than the synchronous case.

In modern large-scale power systems, the UC problem is solved in a high frequency manner throughout the entire day. In order to demonstrate the computational gains in a high frequency environment, we apply the asynchronous method to solve the UC problem over 50 consecutive runs with the same runtime parameters. Figure 3.4(a), a bar graph demonstrates the cumulative time taken by our algorithm with respect to its synchronous counterpart over all the 50 runs thereby implying the tremendous computational advantage presented by the asynchronous algorithm in a high frequency setting.

To measure the quality of solution of the optimization model, we use a quantity known as the *optimality gap*. It is the relative difference in objective values with respect to the optimal solution of the problem which is provided by the centralized method in our case. Therefore, we used the objective value yielded by the centralized solution to calculate our optimality gap.

Figure 3.4(b) shows a scatter plot of the optimality gap yielded by Algorithm 2 versus total convergence time over all the 50 runs in blue while the synchronous result is shown in green. We can see that the optimality gap for the asynchronous algorithm is higher than that of the synchronous although the asynchronous version mostly takes much lesser time to arrive at a tolerable optimality gap. While an optimality gap of zero is an interesting theoretical problem, in practice it is more preferable to obtain a small optimality gap in a much faster way[26]. Further, we can also infer from the figure that the optimality gap has an upper bound at approximately $8\%$. Since our problem is non-convex and cannot be reduced to a consensus optimization problem in its relaxed state, we see a variation in solution quality and a strong correlation between solution time and quality can be inferred. This variation is due to bad solution paths taken by the algorithm which leads to a deteriorating optimality gap.

## 3.6 Conclusion

In this chapter, we present an asynchronous decentralized algorithm for solving the MIP UC problem in power networks. By decomposing the power network into regions with smaller sub-problems, we can decentralize the solution to the UC problem wherein each region solves its locally held UC problem with respect to its network constraints and on the basis of information received from its neighbor regions. In order to handle the complexity stemming from the non-convex nature of our problem and make our solution more robust, we employ a two-stage asynchronous solution where the first phase solves a relaxed version of the UC problem followed by the non-convex MIP version of the same problem. We use a 10 region partition of the IEEE 118 bus problem to demonstrate that our asynchronous decentralized algorithm outperforms its synchronous counterparts in terms of the computation, communication and idle time as well as with respect to the convergence time and the solution quality as assured by the optimality gap. We show that our algorithm is specifically suited to high frequency applications of the UC problem owing to a significantly less time incurred in its overall execution as opposed to the traditional synchronous approach.

From the analysis of the results we can draw a number of conclusions. First, solving non-convex problems asynchronously using ADMM may lead to bad solution paths in some cases which affects the solution quality. Second, there exists a very strong correlation between solution time and quality in such cases. Third, despite the non-convex nature of the problem asynchronous methods are capable of providing significant computational benefit while yielding a solution that is in close approximation to that of synchronous methods.

# CHAPTER 4
# AN ASYNCHRONOUS, DECENTRALIZED SOLUTION FRAMEWORK FOR THE LARGE SCALE UNIT COMMITMENT PROBLEM

## 4.1 Introduction

Any power system optimization framework can be broadly divided into two parts, the data component and the problem component. While the problem component consists of the solution methodology and the optimization problem formulation, the data component consists of infrastructural information pertaining to the network topology, transmission lines, buses and generators. Usually, these problems are solved in a centralized manner at a control center where the problem component must be co-located with the data component.

A distributed UC solution relies mainly on parallel two-stage techniques in order to obtain faster solution times. Such techniques exploit the scope for parallelism among the two stages to provide computational speedup [27, 28, 29, 15]. However they fail to remove the constraint of co-locating the data and the problem components. The UC problem is a stepping stone towards more complex planning problems. Emerging applications of UC involving data-driven operations planning will not be amenable to a centralized model of computation. This will be particularly important as UC problem faces new challenges in data acquisition and interpretations in the areas of integration of renewables, incorporation of maintenance, transmission line switching and prevention of cascading failures. In such problems, co-locating data and problem components may prove to be infeasible.

Unlike the distributed method, a decentralized solution framework is obtained by decomposing the global UC problem component into smaller local subproblems held by multiple computing agents. Each agent only holds a slice of the global data pertaining to its own local subproblem. Therefore, only the local subproblem and the corresponding net-

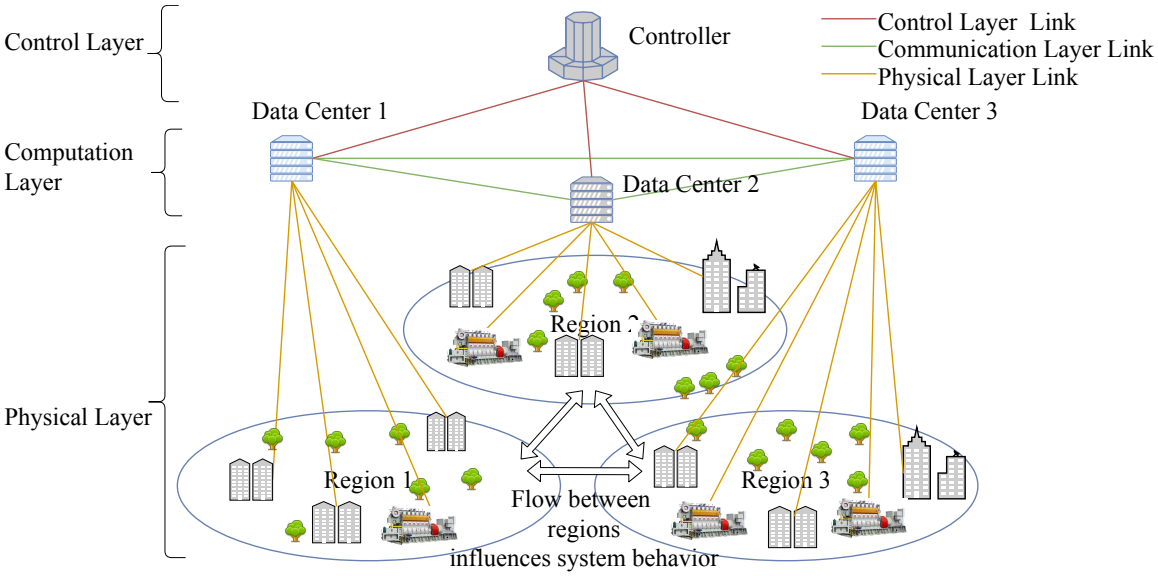



Figure 4.1: A power network schematic for a decentralized framework

work data needs to be co-located. The decentralized UC problem considered in this chapter utilizes a region based decomposition strategy for a large scale power network. Every region is assigned to a unique computing agent denoted by a data center as shown in Figure 4.1. The region based decentralized framework is laterally divided into three parts: physical, computation and control layers. The system level data comprising of network flow estimates is gleaned from the physical network layer by data centers located in the computation layer. These data centers, scattered across a wide geographic area are responsible only for their own local subproblems. At each region, local computation at the data centers yields new estimates of network flow variables that govern transmission lines. The updated estimates pertaining to shared transmission lines between two regions is communicated by the corresponding agents. The newly received network flow estimate is used as input to the next local computation step. The global progress of such an iterative scheme involving local computation and communication is tracked by the control layer. Eventually, the decentralized approach leads to balance of network flow estimates among the regions thereby signifying a solution to the global UC problem.

A decentralized approach to the UC problem has numerous advantages. First, it im-

proves computational efficiency since a region based decomposition of the global problem yields much smaller local subproblems that can be solved in parallel. Second, since each region operates independently, a decentralized method is highly suited for a geographically distributed computational architecture. Third, sensitive operational data can be held privately by each region, thus, limiting data sharing to specific parameters that do not violate privacy. Lastly, it has been found that a decentralized agent-to-agent communication is more efficient in terms of communication latencies [30, 31]. A distributed agent-to-master communication model introduces a single point of failure with the master bearing the heavy burden of processing information sent by all the workers. Since the master can only process information from one agent at a time, the other agents must wait till the master has had an opportunity to process and respond to their corresponding message resulting in poor scalability with respect to increasing problem sizes. Since communication is more expensive than computation [32] idle time can be eliminated or significantly reduced by allowing agents to send messages to each other instead of waiting on one master to respond As a result decentralized communication improves computational performance.

However, one of the disadvantages of current decentralized methods is the need to fully *synchronize* computation among all agents. In synchronous settings, all regions perform their local computational step, and wait for the other regions to finish before proceeding to the communication step. In a large scale decentralized power network, the expectation of a fully synchronous system can be highly misplaced [23]. Local subproblems might vary in their computational complexity owing to different problem sizes, which often leads to computational imbalance. The computing hardware employed by the various regions might be different, leading to an imbalance at the hardware level itself. In reality, a region's UC problem is localized to a data center within close geographical proximity to regional power assets. A practical solution to a large scale decentralized UC problem with a sizable number of regions would involve multiple data centers scattered over a vast geographical area thereby incurring significant communication costs. Therefore, in a real world implementa-

tion, a synchronous approach might suffer from significant idle time incurred due to local heterogeneity and increased communication costs leading to poor computational efficiency and slower progress.

An asynchronous approach has the potential to provide reliable, fast and robust decisions for power system optimization problems. In an asynchronous setting, all the regions perform their local updates based on the latest available information from their peers. Therefore, unlike the synchronous approach, the computational bottleneck arising from slower regions is eliminated, which minimizes idle time and improves computational efficiency. It naturally follows that an asynchronous method would also be resilient to the computational imbalance of local problems due to heterogeneous hardware and communication latencies. Further, an asynchronous decentralized method is regarded more favorably with respect to the mitigation of cyber-attacks since regional computations progress independently and the global objective remains unchanged.

In this chapter, we focus on developing a novel *asynchronous decentralized computational framework* for solving large scale UC problems. The main contributions of this chapter can be summarized as follows:

- We develop a two phase asynchronous decentralized algorithm for solving the decentralized UC problem asynchronously. The algorithm iteratively solves the convex relaxation in the first phase. In the second phase, the binary constraints on the decision variables are imposed. We strengthen our two phase approach with privacy preserving valid inequalities that lead to sound solution quality and robust computational performance.

- We propose a novel *interleaved binary* mechanism that allows regions to advance to the binary phase after having exhibited consistent local convex convergence behavior irrespective of global convex convergence. This also leads to a significant improvement in computational efficiency.

- We propose and implement the concept of a controller to facilitate two-way message exchanges at discrete global clock ticks among neighboring regions which satisfies a crucial convergence assumption [33].

- We develop a custom-made software framework based on the asynchronous reformulation that is fine tuned specifically for the UC problem. We present simulations with respect to the 75, 100 and 120 region scenarios of the IEEE 3012 bus system on a high performance computing environment using MPI semantics.

A typical application of our method would be for a large scale ISO or vertically integrated power company. In such a scenario, participating regions could solve the global problem in a decentralized manner without revealing their infrastructural data while having superior computational performance with respect to centralized methods. Our algorithm is mainly for transmission level operators. However, our methodology is generic and could potentially be applied for coordinating transmission and distribution coordinated system operations as well.

The rest of the chapter is divided as follows. Section 4.2 we discuss various other approaches explored in literature with respect to decentralized, asynchronous methods. In Section 4.3, we present the region based decentralized decomposition of the UC problem. We present the asynchronous decentralized algorithm in Section 4.4 based on [33] and develop a privacy preserving valid inequality that delivers algorithmic improvements. We discuss the implementation aspects of our algorithm and introduce the concept of a controller to successfully orchestrate two-way message exchanges as required by our algorithm. We present results from a robustness and benchmarking study in Section 4.5 and compare the asynchronous method against the centralized and synchronous variants. We present our conclusions in Section 4.6.

## 4.2 Related Works

Augmented Lagrangian techniques like the Alternating Direction Method of Multipliers (ADMM) [12] have been useful for solving decentralized constrained optimization problems with good convergence properties [13]. In the asynchronous optimization literature, the main focus has been on unconstrained optimization [34, 35, 36]. Recently, there has been a growing literature on asynchronous constrained models to address a more general class of optimization problems. Chang [37] proposes an asynchronous ADMM oriented solution for constrained optimization problems with time-varying networks and also under communication errors, whereas Eckstein [38] proposes an asynchronous ADMM like method for multi-block decomposable problems suited for an HPC environment with shared memory capabilities and all-to-all connectivity among compute nodes. In contrast, Wei and Ozdaglar [33] propose an asynchronous ADMM algorithm for distributed constrained optimization that makes no assumptions about compute capabilities of the hardware or about the communication links present in the network. Further the authors show the convergence of their algorithm for distributed constrained optimization problems making their algorithm highly suited for a geographically distributed set of compute agents communicating over potentially high delay inducing links.

Asynchronous master-slave type distributed optimization techniques have recently gained popularity within the power systems domain. Zhang and Kwok [25] propose an ADMM implementation that takes advantage of partial progress being made by slaves with respect to the master, resulting in higher computational efficiency. Within the domain of power systems a similar idea is explored by Aravena and Papavasilou[27] in terms of a distributed asynchronous two stage stochastic UC model where the dual iterations and the feasibility recovery between the master and the slaves occurs asynchronously. Papavasilou *et al.*[15] propose a HPC solution framework for solving the stochastic UC problem with dual decomposition. Similar work has been done in the asynchronous domain by Kim *et al.* [28,

29] exploiting the asynchrony arising out of load imbalance between the various slaves and the master in two-stage stochastic problems specifically for the security constrained UC and the stochastic UC problems. A limitation of these methods over the decentralized approaches, is the requirement of a master node. While offering much potential in terms of computational efficiency, the presence of a master problem with infrastructural data pertaining to the entire network can prove to be impractical in a real world setting owing to reasons mentioned in Section 4.1.

A hallmark of a decentralized solution is the absence of a master problem holding global data. In this domain, the work done by Feizollahi *et al.* [18] provides a synchronous decentralized fix-and-release approach for large scale UC problems. However, in their framework, the UC fix-and-release pertaining to the MIP may not be ideal for a geographically distributed computing environment. A direct asynchronous extension of the synchronous decentralized computational framework for UC presented in [18] has been explored in [39, 40]. In a similar context, Guo *et al.* [41] provide an asynchronous decentralized method for non-convex problems in power systems, which has a similar computational outline as the previous two works but as pointed out in [39], such an approach might suffer from poor solution quality when it comes to MIP problems such as asynchronous decentralized UC. While these methods show great promise in computational efficiency, the solution quality arising from such types of asynchronous decentralized UC frameworks has been shown to be poor at times.

### 4.3 Decentralized Unit-Commitment

In this section, we present an enhanced decentralized UC problem formulation that is derived from the asynchronous ADMM framework for constrained optimization problems proposed by Wei and Ozdaglar [33]. This approach targets a multi-agent decentralized solution to the constrained optimization problem. Each agent exchanges its local estimate of the consensus variable with a neighbor after each local solve. A message exchange be-

tween two neighbors is triggered by the local clock tick associated with that edge. In doing so iteratively, all the agents converge to a solution for the global optimization problem. In our current formulation, we rely on two-way exchange of messages by a pair of neighboring regions at each global tick unlike our previous work [39] that used a broadcast based method. In addition to the consensus quantities themselves, we also exchange their respective Lagrangian information in order to adhere to the convergence conditions set forth in [33].

Our decentralized formulation for the UC problem is geared towards improving solution quality in an asynchronous setting to bolster its applicability in a real world, geographically distributed computational environment. In asynchronous computational conditions, heuristics proposed in the literature with a synchronous approach in mind (i.e. fix-and-release) might be impractical. Further, the solution mechanism illustrated in [39] makes it evident that decentralized asynchronous methods offer good computational potential but leave a lot of room for improvement in terms of the solution quality owing to oscillations from the binary components of the UC problem.

A region based decomposition partitions the set of buses such that each bus is uniquely owned by only one region. It follows that every bus in a region can be categorized either as a boundary or an internal bus. Boundary buses of a region are ones which have a transmission line connecting them to at least one bus belonging to a neighboring region. Further, buses owned by a neighboring region lying on the other end of a transmission line from a boundary bus are termed as foreign buses. On the other hand, all buses owned by a region which have no transmission lines connecting to foreign buses are termed as internal buses. Transmission lines are identified using a unique identifier representing the buses at either end.

We present the decentralized UC objective function in Problem (4.1), the model and the

constraints are listed in (4.2). Details about notations have been provided in Appendix B.

$$\begin{aligned}\hat{\mathcal{L}}_r(\bar{\boldsymbol{\theta}}, \bar{\boldsymbol{F}}, \boldsymbol{\lambda}, \boldsymbol{\phi}) = &\sum_{t\in T}\sum_{g\in G_r} d^g y_t^g + c^g x_t^g + \sum_{t\in T}\sum_{g\in G_r} S_U^g \pi_{Ut}^g + S_D^g \pi_{Dt}^g \\ &+ \sum_{t\in T}\sum_{b\in\mathcal{B}_r}\left[\lambda_t^b|\theta_t^b - \bar{\theta}_t^b| + \frac{\rho_\theta}{2}(\theta_t^b - \bar{\theta}_t^b)^2\right] \\ &+ \sum_{t\in T}\sum_{u\in\mathcal{U}_r}\sum_{v\in\mathcal{V}_r^u}\left[\phi_t^{uv}|f_t^{uv} - \bar{f}_t^{uv}| + \frac{\rho_f}{2}(f_t^{uv} - \bar{f}_t^{uv})^2 + \frac{\rho_f}{2}(f_t^{uv} - \tilde{f}_t^{uv})^2\right]\end{aligned} \tag{4.1}$$

$$\min_{\boldsymbol{\theta},\boldsymbol{f},\boldsymbol{x},\boldsymbol{y}} \ \hat{\mathcal{L}}_r(\bar{\boldsymbol{\theta}}, \bar{\boldsymbol{F}}, \boldsymbol{\lambda}, \boldsymbol{\phi}) \tag{4.2a}$$

$$\text{s.t. } P_{min}^g x_t^g \le y_t^g \le P_{max}^g x_t^g, \quad \forall t \in T, \forall g \in G_r \tag{4.2b}$$

$$-\pi_{Dt}^g \le x_t^g - x_{t-1}^g \le \pi_{Ut}^g, \ \forall t \in [2,T], \forall g \in G_r \tag{4.2c}$$

$$-R^g \le y_t^g - y_{t-1}^g \le R^g, \ \forall t \in [2,T], \forall g \in G_r \tag{4.2d}$$

$$\Gamma^{uv}(\theta_t^u - \theta_t^v) = f_t^{uv}, \ \forall u \in \mathcal{U}_r, \forall v \in \mathcal{V}_r^u, \forall t \in T \tag{4.2e}$$

$$-F_{max}^{uv} \le \Gamma^{uv}(\theta_t^u - \theta_t^v) \le F_{max}^{uv}, \ \forall u \in \mathcal{U}_r \cup \mathcal{I}_r, \quad \forall v \in \mathcal{B}_r^u, \forall t \in T \tag{4.2f}$$

$$\sum_{\forall g\in G_r^u} y_t^g - \delta_t^u = \sum_{\forall v\in\mathcal{B}_r^u}[\Gamma^{uv}(\theta_t^u - \theta_t^v)], \ \forall u \in \mathcal{U}_r \cup \mathcal{I}_r, \quad \forall t \in T \tag{4.2g}$$

$$\sum_{\forall i\in U_t} \pi_{Ui}^g \le x_t^g \le 1 - \sum_{\forall i\in D_t} \pi_{Di}^g, \ \forall t \in T, \forall g \in G_r$$
$$U_t = [t - M_U^g + 1, t], D_t = [t - M_D^g + 1, t] \tag{4.2h}$$

Constraint (4.2b) ensures production at each generator being bounded by its minimum and maximum capacity. Constraints (4.2c) and (4.2h) enforce minimum up and down time for each generator. Constraint (4.2d) ensures generators adhere to their respective ramping

limitations. Equation (4.2e) ensures that flow across inter-regional links is a function of the respective phase angles. Constraint (4.2f) enforces transmission line capacity constraints. Equation (4.2g) ensures that at each bus the demand is met by either power generated by attached generators, or with the flow into the region. Equations (4.2e)-(4.2g) enforce global network flow constraints.

We estimate two important quantities, *intermediate flows* $\bar{F}_t^{uv}$ and *intermediate phase angles* $\bar{\theta}_t^b$ using Equation (4.3) based on values received from neighboring region $r' \in \mathcal{N}_r$. In each update, region $r'$ sends flow dual values $\tilde{\phi}_t^{uv,r'},\ \forall u \in \mathcal{U}_r, \forall v \in \mathcal{V}_r^u \cap \mathcal{U}_{r'}, \forall t \in T$ and phase angle and phase dual estimates $\tilde{\theta}_t^{b,r'}, \tilde{\lambda}_t^{b,r'} \forall b \in \mathcal{B}_r \cap \mathcal{B}_{r'}$.

$$\hat{\lambda}_t^b = \frac{-1}{2}(\lambda_t^b + \tilde{\lambda}_t^{b,r'}) + \frac{\rho_\theta}{2}(\theta_t^b - \tilde{\theta}_t^{b,r'}) \tag{4.3a}$$

$$\bar{\theta}_t^b = \frac{1}{\rho_\theta}(\hat{\lambda}_t^b + \lambda_t^b) + \tilde{\theta}_t^{b,r'}, \quad \lambda_t^b = \hat{\lambda}_t^b \tag{4.3b}$$

$$\hat{\phi}_t^{uv} = \frac{-1}{2}(\phi_t^{uv} + \tilde{\phi}_t^{uv,r'}) + \frac{\rho_f}{2}(f_t^{uv} - \tilde{f}_t^{uv,r'}) \tag{4.3c}$$

$$\bar{f}_t^{uv} = \frac{1}{\rho_f}(\hat{\phi}_t^{uv} + \phi_t^{uv}) + \tilde{f}_t^{uv,r'}, \quad \phi_t^{uv} = \hat{\phi}_t^{uv} \tag{4.3d}$$

In order to avoid redundant and expensive communication, the flow estimates $\tilde{f}_t^{uv}$ of the neighbor region are computed based on the phase angles sent by the neighbor.

## 4.4 Asynchronous Solution Methodology

In this section, we seek to design a solution methodology for Problem (4.2) that performs well in asynchronous conditions and maintains operational privacy. We augment the existing formulation with a redundant valid inequality based on production and demand to boost computational performance in an asynchronous system. The two-way message exchange necessitated by local clock tick as mentioned in [33] and explained in Section 4.3 is tedious to implement and may cause significant computational overhead. Therefore, we introduce the concept of a controller that matches two neighboring regions on the completion of their respective local computation step. We design and develop the controller mechanism to

also track asynchronous global progress of the regions while preserving asynchronous convergence conditions. Finally, we present our two phase asynchronous decentralized UC algorithm that solves the local convex relaxations of Problem (4.2) in the first phase before imposing binary constraints in the second phase. In order to improve computational speedup, our algorithm incorporates a novel interleaved binary mechanism that lets regions advance to their local binary problems based on consistent local convergence of their convex relaxations.

### 4.4.1 A privacy preserving valid inequality

In a decentralized UC solution, global production has a direct bearing on the binary commitment variables. Theoretically, network flow constraints ought to be sufficient conditions for the UC problem solution to balance global production and demand. However, in a decentralized environment, network flow constraints are enforced using Lagrangian decomposition between regions. Decomposed network flow constraints drive the optimal assignment of binary decision variables which ultimately culminates in global convergence. Owing to volatility arising out of heavy latency induced message passing in an asynchronous system, the network flow constraints alone might not be strong enough to meet this balance. The decentralized formulation must therefore be further secured by a globally redundant constraint based on production and demand. These constraints must also retain the privacy preserving nature of decentralized methods.

We consider a balance of global production and demand denoted by $\sum_{\forall r \in \mathcal{R}} \sum_{\forall b \in \mathcal{U}_r \bigcup \mathcal{I}_r} \delta_t^b = \sum_{\forall r \in \mathcal{R}} \sum_{\forall g \in G_r} y_t^g$. In order to decentralize the production-demand balance constraint, we establish an asynchronous friendly mechanism in order to enforce it globally. We respectively compute the local production difference $\psi_{r,t}$, the inverse of the average production residual cost $s_{r,t}$ and its multiplier $\mu_{r,t}$ for every region $r$ and for every time period in the planning

horizon as follows.

$$\psi_{r,t} = \sum_{\forall b \in \mathcal{U}_r \bigcup \mathcal{I}_r} \delta_t^b - \sum_{\forall g \in G_r} y_t^g \tag{4.4a}$$

$$s_{r,t} = \frac{\sum\limits_{\forall g \in G_r} (P_{max}^g - y_t^g)}{\sum\limits_{\forall g \in G_r} d^g (P_{max}^g - y_t^g)} \tag{4.4b}$$

$$\mu_{r,t} = \frac{s_{r,t}}{\sum\limits_{\forall r \in \mathcal{R}} s_{r,t}} \tag{4.4c}$$

Then, the local production target is then computed as follows.

$$\bar{p}_{r,t} = \sum_{\forall g \in G_r} y_t^g + \mu_{r,t} \sum_{\forall r \in \mathcal{R}} \psi_{r,t} \tag{4.5}$$

Therefore, Problem (4.1) is further augmented by Lagrangian penalty terms pertaining to production difference as described in Problem (4.6).

$$\mathcal{L}_r \min_{\boldsymbol{\theta}, \boldsymbol{f}, \boldsymbol{x}, \boldsymbol{y}} \quad \hat{\mathcal{L}}_r(\bar{\boldsymbol{\theta}}, \bar{\boldsymbol{F}}, \boldsymbol{\lambda}, \boldsymbol{\phi}) \tag{4.6a}$$

$$+ \sum_{t \in T} \left[ \eta_t |p_{r,t} - \bar{p}_{r,t}| + \frac{\rho_p}{2} (p_{r,t} - \bar{p}_{r,t})^2 \right]$$

$$\text{s.t.} \quad \text{(4.2b)} - \text{(4.2h)}$$

$$\sum_{\forall g \in G_r} y_t^g = p_{r,t}, \quad \forall t \in T \tag{4.6b}$$

We further add Equation (4.6b) to the existing constraint set where we try to provide a production target for Problem (4.6). This is achieved by performing a global weighted average of the production difference arising out of every region. Intuitively, each region is assigned a customized production target as given by $\bar{p}_{r,t}$ further strengthening convergence. It is important to note that computation of $\bar{p}_{r,t}$ is highly suited for an asynchronous model owing to a reduced volatility in values due to the multiplier $\mu_{r,t}$.

### 4.4.2 Compute Architecture

One of the most important conditions imposed by the asynchronous algorithm in [33] is the presence of a global clock that drives two-way exchange of messages among agents [33]. At each global clock tick a pair of neighboring agents are triggered and exchange local information with each other leading to a two-way message exchange paradigm referred to as a doubly stochastic system.

The requirement of double stochasticity can prove to be a limitation for a variety of reasons. From a practical standpoint, especially in a geographically distributed computational setup, the implementation of a global clock is tedious and leads to a heavier computational burden with lesser accuracy [42]. From a computational perspective, techniques relying on a doubly stochastic system also suffer from issues related to potential bottlenecks in case of compute node failure [43]. If not given designed in the right manner, doubly stochastic systems can undermine the benefits of a decentralized method.

In order to solve a decentralized asynchronous constrained optimization problem, it is necessary to comply with the convergence conditions proposed in [33] and simultaneously address the issues arising out of a doubly stochastic algorithm. Therefore, in this chapter we propose the concept of an additional computational agent called a controller. The controller for a doubly stochastic asynchronous decentralized scheme has the following roles:

- facilitating exchange of messages between neighboring regions on each clock tick.
- helping detect global convergence of the algorithm for an asynchronous method.
- computing an estimate of a global sum asynchronously.

Algorithm 3 introduces the logic behind the asynchronous controller. The controller maintains a running list across the planning horizon of the global production difference vector $\tilde{\boldsymbol{\psi}}^r$, the global inverse average residual cost vector $\tilde{\boldsymbol{s}}^r$, the local convergence values $\tilde{\xi}^r$ and the phase $\tilde{\kappa}^r$ of each region. As soon as a region finishes its local computation,

**Algorithm 3** Controller Logic

initialize $\tilde{\boldsymbol{\psi}}^r, \tilde{\boldsymbol{s}}^r, \tilde{\xi}^r, \tilde{\kappa}^r \leftarrow 0, \forall r \in \mathcal{N}$
**while** GC = false **do**
  recv $\{\tilde{\boldsymbol{\psi}}^{r_1}, \tilde{\boldsymbol{s}}^{r_1}, \tilde{\xi}^{r_1}, \tilde{\kappa}^{r_1}\}$ from some region $r_1$
  **if** $\tilde{\xi}^{r_2} = 1$, such that $\exists r_2 \in \mathcal{N}_{r_1}$ **then**
    send to $r_1$ $\{\sum_{r=1}^{|\mathcal{R}|} \tilde{\boldsymbol{\psi}}^r, \sum_{r=1}^{|\mathcal{R}|} \tilde{\boldsymbol{s}}^r, \sum_{r=1}^{|\mathcal{R}|} \tilde{\xi}^r, r_2\}$
    send to $r_2$ $\{\sum_{r=1}^{|\mathcal{R}|} \tilde{\boldsymbol{\psi}}^r, \sum_{r=1}^{|\mathcal{R}|} \tilde{\boldsymbol{s}}^r, \sum_{r=1}^{|\mathcal{R}|} \tilde{\xi}^r, r_1\}$
  **end if**
  if $\tilde{\xi}^r, \tilde{\kappa}^r = 1, \forall r \in \mathcal{R}$, then GC $\leftarrow$ true
**end while**

it sends its updated local production residual as well as the inverse average residual cost, its local convergence value and its phase to the controller. The controller updates its running estimate of these values and tries to match the aforementioned region with any of its neighbors that has also completed its local computation thereby preserving the concept of a global clock. If a match is found, the latest global running estimates are communicated to the matched pair. If no other neighboring region is active, the region simply waits until it hears back from the controller. GC denotes global convergence which occurs when all regions have converged with respect to the binary phase. Any private infrastructural data of the regions remain opaque to the controller, since production difference and the multiplier values shield any private information local to the regions.

### 4.4.3 Asynchronous Decentralized UC Algorithm

The convex relaxation of the UC problem plays a key role in obtaining good solution quality [18, 39]. It is also clear that UC problems in general have a very good convex relaxation[44]. Therefore, in this chapter, we focus on improving the overall solution quality and computational performance of the asynchronous decentralized UC problem by first strengthening the performance of the convex relaxation with the help of the framework proposed in [33].

We divide the problem into two phases pertaining to the convex phase followed by the

binary phase. In each phase, the Lagrangian values are also exchanged in addition to the phase angles and flow information, as dictated by [33], in order to obtain good solution quality. The solution from the convex phase is then used as a starting point for solving the binary phase.

**Algorithm 4** Interleaved Binary Asynchronous Decentralized UC (ADUC) Algorithm

**for** $r = 1, 2, 3 \ldots |\mathcal{R}|$ **do**
  Initialize $\bar{\boldsymbol{\theta}}_0, \bar{\boldsymbol{f}}_0, \bar{\boldsymbol{F}}_0, \boldsymbol{\lambda}_0, \boldsymbol{\phi}_0, \boldsymbol{x}_0, \boldsymbol{y}_0, k \leftarrow 0$
  $\kappa \leftarrow 0$, set starting phase to convex
  **while** GC = false **do**
    **if** $(||\boldsymbol{\theta}_k - \tilde{\boldsymbol{\theta}}_k|| < \alpha)$ and $(||\tilde{\boldsymbol{\theta}}_k - \tilde{\boldsymbol{\theta}}_{k-1}|| < \beta)$ **then**
      set $\xi_k \leftarrow 1$
      if for last $\zeta$ updates, $\xi_i = 1$, then set $\kappa \leftarrow 1$
    **end if**
    $\boldsymbol{\theta}_{k+1}, \boldsymbol{f}_{k+1}, \boldsymbol{x}_{k+1}, \boldsymbol{y}_{k+1}$ calculated by Problem (4.6)
    calculate $\boldsymbol{\psi}, \boldsymbol{s}, \boldsymbol{\mu}$ using Equation (4.4)
    send $\{\boldsymbol{\psi}, \boldsymbol{s}, \xi_k, \kappa\}$ to the controller
    recv $\{\sum_{r=1}^{|\mathcal{R}|} \tilde{\boldsymbol{\psi}}^r, \sum_{r=1}^{|\mathcal{R}|} \tilde{\boldsymbol{s}}^r, \sum_{r=1}^{|\mathcal{R}|} \xi^r, r'\}$ from controller
    **if** $\sum_{r=1}^{|\mathcal{R}|} \xi_r = |\mathcal{R}|$ **then**
      if $\kappa = 1$, set GC $\leftarrow$ true, otherwise $\kappa \leftarrow 1$
    **end if**
    compute $\bar{\boldsymbol{p}}_r$ using Equation (4.5)
    send tuple $\Delta_{rr'}^{ADUC} = \{\boldsymbol{\Theta}, \boldsymbol{\Lambda}, \boldsymbol{\Phi}\}$ to $r'$
    recv tuple $\Delta_{r'r}^{ADUC} = \{\tilde{\boldsymbol{\Theta}}_{r'}, \tilde{\boldsymbol{\Lambda}}_{r'}, \tilde{\boldsymbol{\Phi}}_{r'}\}$ from $r'$
    compute $\tilde{\boldsymbol{f}}_{r'}$ based on $\tilde{\boldsymbol{\Theta}}_{r'}$
    update $\bar{\boldsymbol{\theta}}_{k+1}, \boldsymbol{\lambda}_{k+1}, \bar{\boldsymbol{f}}_{k+1}, \boldsymbol{\phi}_{k+1}$, using Equation (4.3)
    $\eta_r^t = \eta_r^t + \rho_p(p_{r,t} - \bar{p}_t), \quad \forall t \in T$
    $k \leftarrow k + 1$
  **end while**
**end for**

The Interleaved Binary Asynchronous Decentralized Unit Commitment (IBAD-UC) solution methodology is presented in Algorithm 4. At every iteration k, each region solves its own local subproblem and generates the commitment decisions $\boldsymbol{x}_k$, dispatch decisions $\boldsymbol{y}_k$, local phase angle values $\boldsymbol{\theta}_k$, flow values $\boldsymbol{f}_k$. The region communicates the updated local production difference values and the inverse average residual cost to the controller and waits for a response. A response from the controller provides an estimate of the global

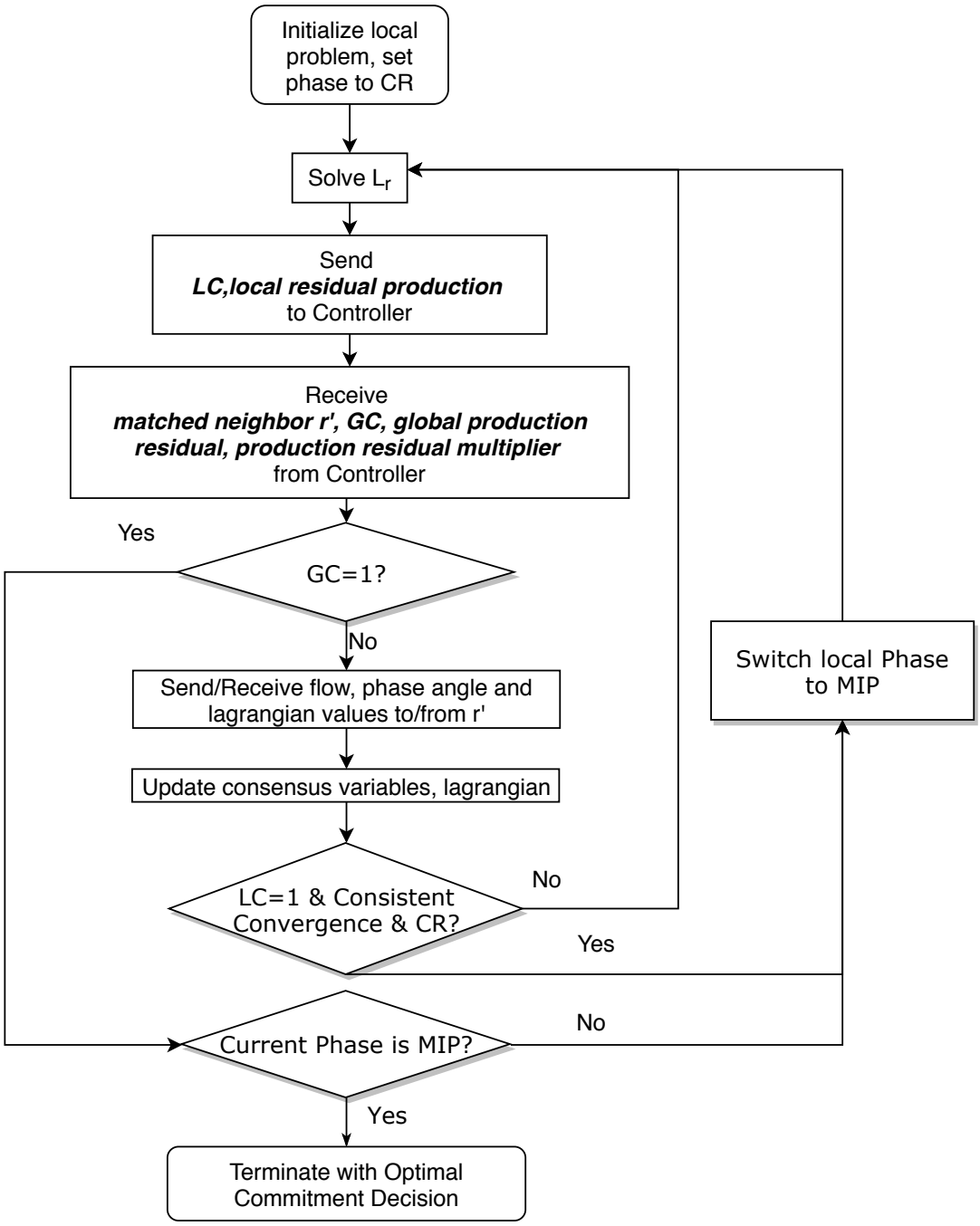



Figure 4.2: Flowchart representing Algorithm 4.

production and the multiplier vector $\sum_{r=1}^{|\mathcal{R}|} \tilde{\psi}^r$, $\sum_{r=1}^{|\mathcal{R}|} \tilde{s}^r$ which are used in the next iteration of the local subproblem. The controller also identifies the neighbor with which the region must perform a two-way message exchange. This step initiates an exchange of information

denoted by the tuple $\Delta_{rr'}^{ADUC}$ between region $r$ and its neighbor $r'$, where,

$$\Delta_{rr'}^{ADUC} = \left\{ \begin{array}{l} \{\boldsymbol{\Theta}_r, \boldsymbol{\Lambda}_r\} = \{\{\boldsymbol{\theta}^b, \boldsymbol{\lambda}^b\} | \forall b \in \mathcal{B}_r \cap \mathcal{B}_{r'}\} \\ \boldsymbol{\Phi}_r = \{\boldsymbol{\phi}^{uv} | \forall u \in \mathcal{U}_r, \forall v \in \mathcal{V}_r^u \cap \mathcal{U}_{r'}\} \end{array} \right\}$$

This tuple consists of the newly generated primal values as well as the Lagrangian information. After observing consistent local convex convergence behavior indicated by the Interleaved Binary (IB) constant $\zeta$, the local subproblem of the region switches from the convex relaxation to its binary counterpart.

Figure 4.2 illustrates the flowchart corresponding to Algorithm 4. $LC$ refers to local convergence value ($\xi$). The convex relaxation phase is denoted by CR ($\kappa = 0$), whereas the imposition of binary constraints on commitment variables is represented by the MIP phase ($\kappa = 1$). Local convergence occurs when the primal and dual variables with respect to the phase angles are close to some predetermined limit denoted by $\alpha$ and $\beta$ respectively. Global convergence occurs when all regions have locally converged with respect to the MIP phase.

## 4.5 Experimental Results

We perform benchmarking studies comparing the IBAD-UC algorithm with its synchronous counterpart to demonstrate its superior computational efficiency and speed. We consider the centralized solution method in which the entire large scale UC problem is solved without region based decomposition. We benchmark the IBAD-UC algorithm with the centralized method to demonstrate comparable solution quality with a significant reduction in solution times.

In order to demonstrate robustness of our proposed solution methodology, we show that the IBAD-UC algorithm yields consistently good quality results with respect to the 75, 100 and 120 region decompositions of the IEEE 3012 bus case. The region decompositions each consist of approximately 40, 30 and 25 buses per region on an average respectively,

depicting a valid real world scenario. We consider 150 generators in the 3012 bus case that have a non trivial production capacity.

### 4.5.1 Experiment Setup

We develop a distributed, parallel software framework that uses the MPI to orchestrate Algorithm 4 on a high performance compute cluster. Within MPI, we rely on Remote Memory Access (RMA) paradigm for asynchronous communication. RMA windows allow remote processes to read and write their latest values. Since each region occupies one process, regions communicate with their neighbors with much simpler semantics offered by RMA.

We perform our experiments on a HPC cluster comprised of Intel Xeon compute nodes with 20 cores per node with a clock rate of 2.80GHz. Each region and the controller are assigned to one core. The *mpi4py* [19] framework was used to interface with MPI to conduct our HPC simulations. *Gurobi 6.5* was used to solve Problem (4.2) locally on each core with multi threading turned off on each region to prevent oversubscription of computational resources. We used IEEE 3012 bus case data from *MATPOWER* [22] for our experiments. Our experimental results pertain to the 24 hour planning horizon spanning an entire day collected from over 400 experiments conducted with the IBAD-UC algorithm. We benchmark our results against the decentralized synchronous solution methodology presented in [18] augmented with the production difference valid inequality with the multiplier being constant $\mu_{r,t} = \frac{1}{|\mathcal{R}|}$.

### 4.5.2 Total solve time and effect of IB Limit

Figure 4.3 presents box plots for the time taken for convergence by the IBAD-UC algorithm against the time taken by its synchronous counterpart. Results depict the effects of variation in the IB Limit parameter represented by $\zeta$. We observe a large variation in the time taken for convergence in the 75 region case with varying $\zeta$, whereas, for 100 and 120 region cases, the variation consistently decreases, with 120 region case being the fastest.

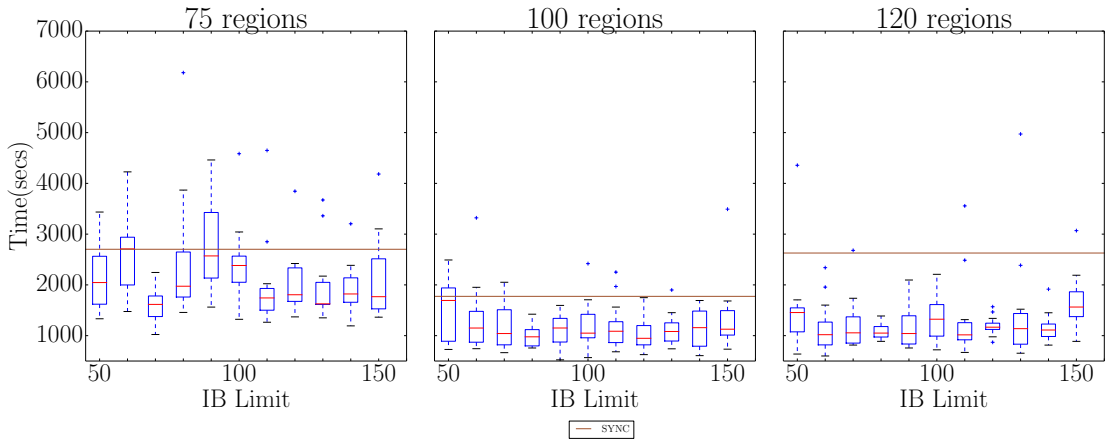



Figure 4.3: Total time taken and the effect of IB limit $\zeta$

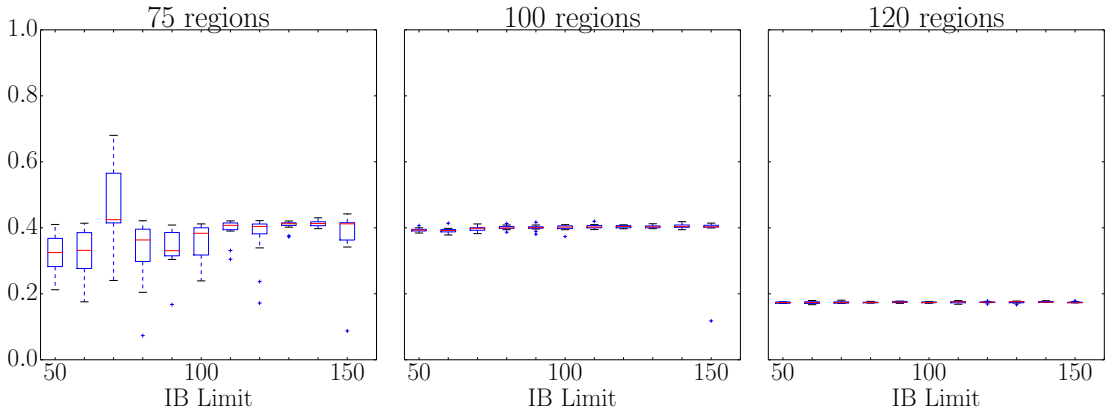



Figure 4.4: Effect of IB limit $\zeta$ on asynchronous degree

Figure 4.3 tells us that the 75 region decomposition likely has a relatively greater degree of imbalance in the problem sizes compared to the 100 and 120 region cases. It is also interesting to note that performance trends with respect to $\zeta$ oscillate between high and low variations in convergence time successively as $\zeta$ is increased, although this trend becomes much more subtle as we move from the 75 to the 120 region case. Overall, it can be seen that despite a variation in performance with respect to $\zeta$, the IBAD-UC algorithm outperforms synchronous in all three cases.

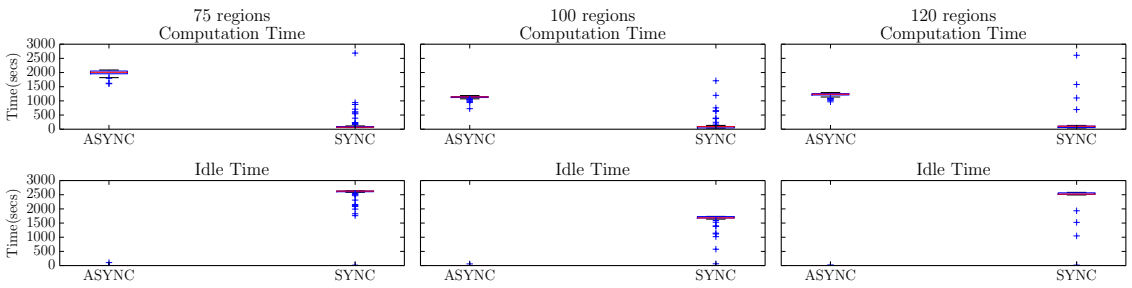



Figure 4.5: Variation in Computation and Idle time for 75,100 and 120 regions

### 4.5.3 Asynchronous Degree

The asynchronous degree is the ratio of the minimum and the maximum number of updates performed by any region within each case. In Figure 4.4 we present box plots of the asynchronous degree based on $\zeta$ as a means to measure how asynchronous the system is. We can see that the asynchronous degree shows significant variation with a lower $\zeta$ value and stabilizes as we increase $\zeta$. For the 100 and 120 region case presented in Figure there is little variation in the asynchronous degree as $\zeta$ is changed. This observation indicates that the 75 region case is relatively imbalanced leading to variation in asynchronous degree which is exacerbated at lower $\zeta$ values indicating premature advancement into the binary phase leading to higher volatility in solution times. Figure 4.4 thereby corroborates Figure 4.3 since higher variation in asynchronous degree might lead to higher variations in solution times as well.

Table 4.1: Timing Analysis (in secs) for 75 Region Case

| | Asynchronous | | Synchronous | |
|---|---|---|---|---|
| Time | Mean | % of Total | Mean | % of Total |
| Comp | 1989.25 | 92.76 | 171.47 | 6.35 |
| Comm | 1.05 | 0.05 | 0.87 | 0.03 |
| Idle | 153.62 | 7.16 | 2528.97 | 93.61 |
| Total | 2144.43 | - | 2701.43 | - |

Table 4.2: Timing Analysis (in secs) for 100 Region Case

| Time | Asynchronous | | Synchronous | |
|---|---|---|---|---|
| | Mean | % of Total | Mean | % of Total |
| Comp | 1128.81 | 94.28 | 130.61 | 7.36 |
| Comm | 0.76 | 0.06 | 0.65 | 0.04 |
| Idle | 67.42 | 5.63 | 1642.65 | 92.58 |
| Total | 1197.32 | - | 1774.23 | - |

Table 4.3: Timing Analysis (in secs) for 120 Region Case

| Time | Asynchronous | | Synchronous | |
|---|---|---|---|---|
| | Mean | % of Total | Mean | % of Total |
| Comp | 1222.36 | 94.11 | 137.03 | 5.21 |
| Comm | 1.14 | 0.09 | 0.87 | 0.03 |
| Idle | 74.87 | 5.76 | 2489.31 | 94.73 |
| Total | 1298.80 | - | 1774.23 | - |

### 4.5.4 Average Times

Tables 4.1,4.2 and 4.3 present the average computation, communication and idle times incurred by the asynchronous and the synchronous methods for the 75, 100 and 120 region cases. Figure 4.5 presents the variation in the mean computation and idle times for every region incurred by IBAD-UC alongside those for the synchronous method.

From the tables, we observe that the average synchronous computation time in general is much lower than the asynchronous while also incurring a smaller percentage share of the total as well. While the asynchronous method spends relatively less time idling, the synchronous method suffers from greater amount of idle time, both in terms of average and the percentage times. In addition, the communication times and the respective percentages do not depict much variation between asynchronous and synchronous methods. The consistency observed in terms of computation, communication and idle times by various region decompositions show robustness in computational performance by the IBAD-UC algorithm.

Figure 4.5 provides deeper insight into the trends presented in Tables 4.1,4.2 and 4.3 by presenting *regional variations* in computation and idle times. We observe that there is

Table 4.4: Centralized Solution

| | |
|---|---|
| Total Objective ($\gamma_c$) | 108316.6 |
| Lower Bound ($\lfloor\gamma_c\rfloor$) | 108218.5 |
| Time (secs) | 19139 |

wide variation in computation and idle times for regions in the synchronous method. For the synchronous method, the highest computation time incurred by a region is also very similar in value to the highest idle time incurred by any region for all cases. However, in the asynchronous method, the lower and upper bounds on computation times are much tighter and idle times are negligible. This behavior in computation and idle time is observed uniformly across all the region decompositions.

The data presented in Figure 4.5 and Tables 4.1,4.2 and 4.3 indicates that the higher computation time for a few regions form the main bottlenecks for global progress which are readily circumvented by asynchronous methods. Meanwhile, global computational progress in the synchronous method is held up by the slowest region which simultaneously incurs very high idling times on the fastest region. Despite strongly asynchronous systems, more frequent asynchronous updates are able to successfully drive the problem towards the global solution much faster leading to superior computational efficiency.

### 4.5.5 Solution Quality

We solve the centralized problem with a 0.1% MIPGAP, the lower bound of which is used to compute the worst case benchmarking for the IBAD-UC algorithm solution quality. We denote $\gamma$ to be the total optimal objective value comprised of operations and commitment components. $\gamma_{async}, \gamma_c$ represent the optimal objective for the asynchronous method and the centralized method respectively. We present the centralized results in Table 4.4 where $\gamma_c$ and $\lfloor\gamma_c\rfloor$ represent the associated objective and lower bound. We use this to calculate a conservative solution quality in terms of the optimality gap($\frac{(\gamma_{async}-\lfloor\gamma_c\rfloor)*100}{\lfloor\gamma_c\rfloor}$) in order to provide the worst case optimality gap for our algorithm. We compute the mean optimality gap among multiple runs of the asynchronous method for 75, 100 and 120 region cases.

Table 4.5: Solution Quality

| Regions | Mean Gap (%) | Std Dev. |
|---|---|---|
| 75 | 1.891 | 0.881 |
| 100 | 1.305 | 0.285 |
| 120 | 1.611 | 0.122 |

Table 4.4 shows that, the asynchronous solution quality is highly consistent among the regions. The asynchronous method on an average is able to consistently solve the decentralized UC problem with less than 2% optimality gap. Drawing insights from Figure 4.4 and Figure 4.3, it can be argued that despite a highly asynchronous system, the variation in solution times as well as the solution quality are relatively small. Further, the optimal objective costs are close to that of the centralized solution indicating a robust solution with higher computational efficiency.

### 4.5.6 Objective Costs

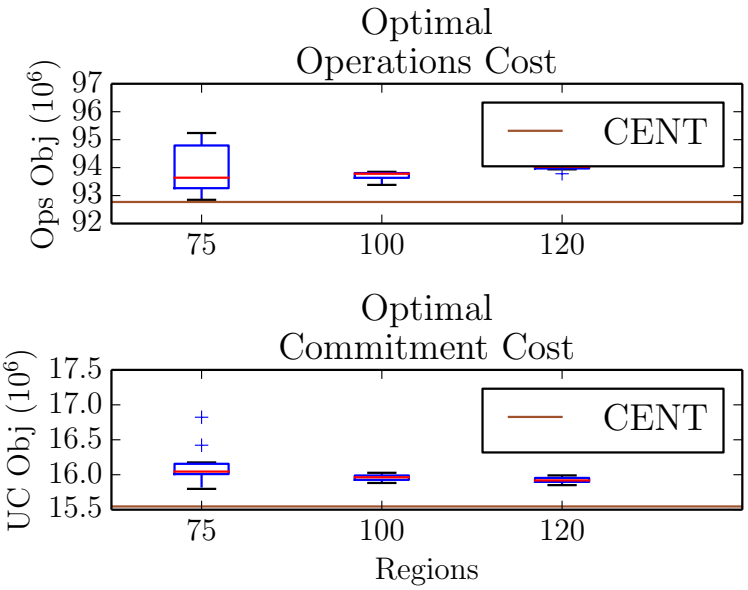



Figure 4.6: Objective costs

Figure 4.6 shows the objective cost comparison of IBAD-UC relative to the centralized solution with respect to the 75,100 and 120 region cases. We can see that the IBAD-

UC yields commitment and operation decisions which are close enough to the centralized method. We can also observe the stability of the solutions within each region case owing to the valid inequalities enforced by IBAD-UC.

## 4.6 Conclusion

In this chapter we present a novel asynchronous decentralized solution methodology for solving the UC problem for large scale power systems. Unlike other asynchronous reformulations proposed in the past that leverage a master-slave hierarchical computational model, our IBAD-UC algorithm is decentralized in nature and intended for a real-time geographically distributed heterogeneous computing environment. Our decentralized problem formulation is constructed with a strong emphasis on the privacy of region level infrastructure data and incorporates a redundant privacy preserving valid inequality. Leveraging the valid inequality the proposed asynchronous method is able to offer considerable algorithmic improvements with respect to stability and robustness of the solution. We propose a controller mechanism that implements two-way message exchanges between regions at discrete global clock ticks without a significant computational burden.

We present HPC simulation studies based on a custom-made software framework developed by us to show the superior computational performance of the asynchronous method, along with stable solution quality. We also benchmark the asynchronous convergence characteristics with respect to the synchronous method and analyze the solution quality against that of the centralized method. Our experiments show that asynchronous methods offer a viable, robust and computationally efficient alternative to the state of the art synchronous decentralized methods.

# CHAPTER 5
# A DECENTRALIZED SUBGRADIENT APPROACH TO THE LARGE-SCALE MAINTENANCE AND UNIT COMMITMENT

## 5.1 Introduction

Power network operations has traditionally been represented by the well-studied Unit Commitment (UC) problem[9]. The UC problem optimizes generator commitment and production decisions subject to network topology, transmission, and generation constraints. Maintenance of generation assets like generators and turbines has received some attention in the literature [45, 46]. Most of the research in this area considered generation maintenance in the context of periodic/calendar-based schemes with much of the developments focusing on optimizing maintenance intervals to minimize cost [46]. There have been some modeling efforts that have attempted to integrate maintenance with network operations. Some of noteworthy examples include [47, 48, 49]. With the exception of [3, 6] most of these models still focus on periodic maintenance. Today, inexpensive sensing technologies have given rise to digital frameworks like the Internet-of-Things (IoT) that is transforming the power generation industry. Predictive analytics based on IoT data are increasingly being utilized to provide advance warnings of impending failures and improve maintenance scheduling. If properly formulated, insights from predictive analytics can be utilized not only to reduce maintenance costs but also to better optimize network operations. More recently, [3, 6] studied the integration of condition based maintenance (CBM) with operations and demonstrated significant cost savings by incorporating sensor-based predictive analytics.

Recent works have focused on decentralized solution methodologies as a means to improve computational performance of large scale mixed integer planning problems like unit commitment [18, 39, 50, 51]. These solution methodologies rely on decomposing the

power network topology into multiple autonomous regions. Consequently, these methods iteratively apply ADMM by dualizing network flow constraints corresponding to transmission lines between regions. However, for solving the joint optimization problem, applying the conventional, ADMM-based decentralization alone is not sufficient for numerous reasons. First, owing to longer maintenance planning horizons, the regional subproblem itself becomes large scale. As a result, the conventional decentralization strategy leads to computationally bulky regional subproblems with significantly higher solution times. Second, the computational issues of long planning horizons is compounded when considering the coupling constraints between maintenance and unit commitment that exists in the joint problem. Consequently, owing to longer planning horizons as well as coupling constraints, the joint maintenance problem formulation is several orders of magnitude more complex to solve than the traditional UC formulation. Lastly, the performance boost of conventional decentralization schemes emanates solely from regional decompositions of the network. Such schemes do not have any bearing on the performance of the local subproblems themselves [50]. Therefore, while decentralization definitely helps, its benefit is marginal when considering the various distinguishing features of the joint problem.

In this chapter, we develop a decentralized framework that is specifically geared towards solving the joint maintenance and unit commitment problem. The hallmark of our approach is that its based on a temporal decomposition of every local subproblem along the maintenance planning horizon. We propose a novel subgradient method based scheme in order to enable this temporal decomposition in addition to conventional, ADMM-based decentralization. Using our approach, we can effectively fan out every local subproblem across multiple threads while still reaping the benefits of conventional decentralized schemes. In fact, the multithreading aspect of our approach compounds the computational benefits of conventional decentralization leading to up to 80% reduction in compute time compared to centralized benchmarks in some cases. Therefore, our framework allows for significantly faster computation of jointly informed, sensor driven maintenance and unit commitment

schedules which would not have been possible if relying on a conventional ADMM-based, decentralized solution paradigm alone.

In this chapter, we propose a novel decentralized framework that addresses the scalability and privacy issues pertaining to the joint CBM and operations optimization problem. As a first step towards scalability, we employ temporal and regional decomposition, and solve each regional problem by dualizing the maintenance cardinality as well as network flow constraints. The regional decomposition is solved in a decentralized fashion while the temporal decomposition is leveraged for multithreading within every regional subproblem. The temporal decomposition driven multithreading, is in turn powered by the subgradient method [52] which effectively constitutes one local solve of the regional subproblem. As a result of the multithreading, the local subproblems become significantly less challenging to solve from a computational standpoint. Next, we iteratively employ the Alternating Direction Method of Multipliers(ADMM) [12] to balance power flow between neighboring regions by dualizing the flow constraints corresponding to respective tie lines. Consequently, the scalability benefits of decentralized methods coupled with computational efficiency from local multithreaded solves make our approach computationally sound. Our contributions in this chapter are summarized as follows:

- We develop a computationally efficient, multithreaded subgradient method by exploiting special structures in the joint formulation in order to solve the local problem.
- We integrate our multithreaded subgradient method as a local solver for the regional subproblems stemming from a conventional ADMM-based decentralized framework.
- We devise a model decomposition and coordination scheme that eliminates the need to move any sensor data used for maintenance or the infrastructure data present at every region. This respects data privacy and significantly reduces cybersecurity threats that can arise from leaked IoT data.
- For evaluation purposes, we develop a High Performance Computing (HPC) imple-

mentation of our decentralized framework based on a hybrid Message Passing Interface (MPI) and OpenMP frameworks. We assign every region to an MPI process and multiple OpenMP threads to every region to achieve a scalable and computationally efficient implementation.

In Section 5.2, we review relevant literature pertaining to decentralized optimization of power systems. Details of the decentralized CBM problem formulation are discussed in Section 5.3. In Section 5.4, we develop the decentralized joint CBM and operations algorithm that preserves regional data privacy. In Section 5.5, we present our experimental results followed by conclusion and future work in Section 5.6.

## 5.2 Related work

Large scale problems in power systems are characterized by long planning horizons and large network sizes. While distributed and decentralized algorithms have both been applied towards solving large scale power system problems, there is a subtle but profound difference between the two categories. Distributed algorithms like those based on cutting planes are typically characterized by the presence of a master process that coordinates the entire computational progress [39]. On the other hand, decentralized algorithms, while being parallel and distributed in nature, function even in the absence of a master process [50]. As a result, decentralized approaches yield two important benefits that are not possible otherwise in generic distributed approaches relying on a master process. Due to the presence only of peer-to-peer message exchanges, decentralized approaches exhibit greater scalability [51]. Further, unlike their distributed counterparts, decentralized methods are usually based on a topology decomposition of the power network into multiple regions that are representative of utilities or their subsidiaries thereof [18]. Each region retains full ownership and control of local data and only exchanges flow information pertaining to shared transmission lines with neighbor regions [18, 50].

Another important differentiator between generic distributed algorithms and decen-

tralized approaches is concerning the implementation aspect. As a consequence of their reliance only on peer-to-peer message exchanges, a decentralized algorithm can be implemented on a geographically distributed computing architecture [39, 50]. On the other hand, decompositions in distributed algorithms are designed mainly for computational convenience with the intent of unleashing massive parallelization potential [15, 17]. Typically, subproblems in distributed algorithms emanate from temporal decompositions for deterministic optimization cases [6] and scenario driven decompositions for stochastic cases [15, 17] without representing any real world entity. Therefore, such approaches can only function on a specialized computational system like an HPC cluster that offers significant parallelization potential but cannot handle geographically distributed problem instances [39]. Owing to such tight limitation of the implementation aspect, distributed approaches force end users to transfer relevant information to an HPC cluster and therefore cannot guarantee privacy of data [51].

Distributed solutions pertaining to operations have conventionally revolved around stochastic UC [15, 17] with scenario based decompositions. The work done in [15] solves stochastic unit commitment problem for a fleet of sustainable energy sources where the imposed demand is time varying. The authors in [17] solve a unit commitment problem using an incremental subgradient method to progress the dual variable while simultaneously recovering primal feasibility. Lately, decentralized approaches for solving deterministic UC are gaining popularity for their enhanced scalability and privacy preserving aspects as stated above [18, 39, 50, 51].

On the other hand, as mentioned in Section 5.1, efforts that have focused on integrating maintenance and network operations are limited. Pertinent examples include [47, 48, 49] where models for optimizing networks operations integrated periodic maintenance of assets and transmission lines. In [53], the authors integrated reliability information, namely generator failure rates, in a simplified operations problem. More recently, [3, 6] have proposed the joint optimization of CBM with operations over a long horizon. Most of these

maintenance solutions have included cutting plane techniques that can be easily implemented in a distributed setting as well [47, 48, 49, 6]. However, techniques to solve the joint maintenance problem in a decentralized fashion have largely remained understated.

## 5.3 Problem Formulation

We propose a decentralized formulation based on regional decomposition with multiple regional subproblems. From a practical standpoint, each region may denote a subsidiary of the utility company in a vertically integrated or a deregulated market. Consider a sample network depicted in Figure 5.1 consisting of 3 regions with boundary and foreign bus categorization for each region defined as follows:

- Region 1: $\mathcal{U}_1 = \{B, C\}$, $\mathcal{V}_1 = \{G, E\}$
- Region 2: $\mathcal{U}_2 = \{E\}$, $\mathcal{V}_2 = \{F, C\}$
- Region 3: $\mathcal{U}_3 = \{G, F\}$, $\mathcal{V}_3 = \{B, E\}$

Every region is comprised of local (critical) generators and buses subject to its own operational constraints. We assume that every critical generator is instrumented with sensors for monitoring health/degradation. Sensor data in each region is streamed to respective local databases where predictive analytic algorithms are used to predict the remaining operational life of the generator. As mentioned earlier, the focus of this chapter is not to develop accurate predictive degradation models but rather to address the computational challenges in large scale joint optimization of operations and maintenance in the presence of IoT-enabled generation assets.

### 5.3.1 Degradation-based Predictive Modeling

Predictive analytics is a crucial aspect of our approach. Degradation-based sensor data from IoT-enabled generators are incorporated into the regional operations and maintenance

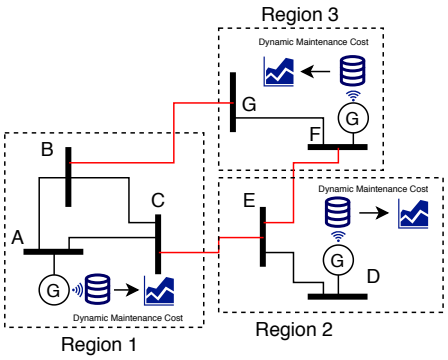



Figure 5.1: Partition of Network topology into regions with full data privacy.

scheduling problems. We leverage contemporary degradation models developed by [10] to compute predictions of generator remaining lifetimes. The authors in [10] proposed a Bayesian framework that utilizes real-time degradation signals from partially degraded assets to predict and continuously update residual life distributions (RLDs). This framework was later extended in [54] by computing optimal replacement/maintenance and spare parts ordering policies driven by real-time RLD predictions. The authors used the RLDs to calculate a convex maintenance cost function to identify the optimal time to perform maintenance. The cost function represents the trade-off between the cost associated with the risk of unexpected failure and the opportunity cost associated with performing premature (or unnecessary) preventive maintenance. As new sensor data is observed, an asset's RLD and the corresponding cost function are updated through a Bayesian framework. [3, 6] adopted this approach in their joint optimization problem and successfully integrated these dynamically evolving expected maintenance cost functions. The expected maintenance cost function at time $t$ is expressed as follows:

$$\omega^g_{t^o,t} = \kappa \cdot \frac{\omega^g_p P(\tau_g > t) + \omega^g_f P(\tau_g \leq t)}{\int_0^t P(\tau_g > z)dz + t_o} \tag{5.1}$$

In Equation (5.1), $\omega^g_{t^o,t}$ is maintenance cost for generator $g$ of age $t^o$ at time $t$. $\kappa$ represents the maintenance criticality coefficient, denoting the relative importance of maintenance with respect to operations. $\omega^g_p$ and $\omega^g_f$ are the costs for preventive maintenance, and unex-

pected failure for generator $g$, respectively while $\tau_g$ is the remaining life of generator $g$. In other words, the function $\omega^g_{t^o,t}$ translates $\tau_g$ (the RLD of generator $g$) into a degradation-based maintenance cost function over time.

### 5.3.2 Decentralized Joint Maintenance and Operations

The regional subproblem seeks to minimize the objective cost as represented by Problem (5.2). For simplicity we consider the vector form of the variables as necessary. In our problem formulation, we let $\mathcal{R}$ denote the set of all regions. Let $\mathcal{N}_r, G_r, \mathcal{U}_r, \mathcal{V}_r, \mathcal{I}_r$ represent the set of neighboring regions, generators, boundary, foreign and internal buses of region $r$. $\mathcal{B}_r = \mathcal{U}_r \cup \mathcal{V}_r$ represents set of all boundary and foreign buses of region $r$. Similarly, $G^b_r, \mathcal{U}^b_r, \mathcal{V}^b_r, \mathcal{I}^b_r$ respectively denote the set of all generators, boundary, foreign and internal buses connected to bus $b \in \mathcal{U}_r \cup \mathcal{I}_r$. The set of all neighboring buses of bus $b$ is denoted by $\mathcal{B}^b_r = \mathcal{U}^b_r \cup \mathcal{V}^b_r \cup \mathcal{I}^b_r$. We let $d^g, c^g, S^g_U, S^g_D$ denote the dispatch cost, commitment cost, start-up cost and shutdown cost of generator $g$ respectively. $\nu$ represents the demand curtailment cost for the network. $P^g_{min}, P^g_{max}$ denotes the minimum and maximum capacity of generator $g$, whereas $\mu^g_U, \mu^g_D, R^g$ represents the minimum up-time, down-time and ramp-up/down constant for generator $g$. $F^{uv}_{max}$ represents the maximum capacity of the line connecting buses $u$ and $v$ such that $u \in \mathcal{U}_r$ and $v \in \mathcal{V}^u_r$. $\Gamma^{uv}$ denotes the phase angle conversion for line $uv$. $\rho_\theta, \rho_f, \rho_p$ respectively represent penalty parameter of the problem for phase angles, flow and load violation terms. $\delta^b_t, \psi^b_t$ represents the demand and the curtailment at bus $b$ at time $t$ respectively. We denote the entire operations planning horizon by the set $T$. We denote $M$ to be the maintenance planning horizon set comprising of multiple smaller operational planning horizons $T_m$ such that,

$$T = \bigcup_{m=1}^{|M|} T_m, \text{ where, } T_m = \left\{ t | t \in \left[ \frac{m|T|}{|M|} \dots \frac{(m+1)|T|}{|M|} \right] \right\}$$

We have $y_t^g, \pi_{Ut}^g, \pi_{Dt}^g \in \mathbb{R}$,$x_t^g, z_m^g \in \{0,1\}$ which represents the production, startup, shut-down, commitment and maintenance variables respectively of generator $g$ at operational time epoch $t \in T_m$. The overall regional production is denoted by $p_{r,t} = \sum_{\forall g \in G_r} y_t^g$. While $\theta_t^b \in \mathbb{R}$ represents the phase angle of bus $b \in \mathcal{U}_r \bigcup \mathcal{V}_r$, $f_t^{uv} \in \mathbb{R}$ denotes power flow from bus $u$ to $v$ where $u \in \mathcal{U}_r$ and $v \in \mathcal{V}_r^u$. Further, $\lambda_t^b, \phi_t^{uv}, \eta_{r,t}$ are the Lagrangian multipliers with respect to phase angles, flow and production difference, where $b \in \mathcal{U}_r \bigcup \mathcal{V}_r$, $u \in \mathcal{U}_r$ and $v \in \mathcal{V}_r^u$.

$$\begin{aligned}\mathcal{L}_r(\Delta) = &\sum_{t \in T} \sum_{g \in G_r} \Big[ c_t^g x_t^g + d_t^g y_t^g + S_U^g \pi_{Ut}^g + S_D^g \pi_{Dt}^g \Big] \\ &+ \sum_{t \in T} \Big[ E_t(\bar{\boldsymbol{\theta}}, \boldsymbol{\phi}) + H_t(\bar{\boldsymbol{F}}, \boldsymbol{\lambda}) + I_t(\bar{\boldsymbol{p}}, \boldsymbol{\eta}) \Big] \\ &+ \nu \sum_{t \in T} \sum_{u \in \mathcal{U}_r \cup \mathcal{I}_r} \psi_t^u + \sum_{g \in G_r} \sum_{m \in M} \omega_m^g z_m^g \end{aligned} \tag{5.2}$$

where $\Delta = \{\bar{\boldsymbol{\theta}}, \bar{\boldsymbol{F}}, \bar{\boldsymbol{p}}, \boldsymbol{\lambda}, \boldsymbol{\phi}, \boldsymbol{\eta}\}$ represents the consensus phase angle, flow, regional production target and their respective lagrangian multipliers. Further,

$$E_t(\bar{\boldsymbol{\theta}}, \boldsymbol{\lambda}) = \sum_{b \in \mathcal{B}_r} \Big[ \lambda_t^b |\theta_t^b - \bar{\theta}_t^b| + \frac{\rho_\theta}{2} (\theta_t^b - \bar{\theta}_t^b)^2 \Big]$$

$$H_t(\bar{\boldsymbol{F}}, \boldsymbol{\phi}) = \sum_{u \in \mathcal{U}_r} \sum_{v \in \mathcal{V}_r^u} \Big[ \phi_t^{uv} |F_t^{uv} - \bar{F}_{t,k}^{uv}| + \frac{\rho_f}{2} (F_t^{uv} - \bar{F}_{t,k}^{uv})^2 \Big]$$

$$I_t(\bar{\boldsymbol{p}}, \boldsymbol{\eta}) = \eta_{r,t}(p_{r,t} - \bar{p}_{r,t}) + \frac{\rho_p}{2} (p_{r,t} - \bar{p}_{r,t})$$

The objective function represented by Problem (5.2), consists of a commitment, operations and maintenance cost components. In addition, it also consists of ADMM penalty terms imposed on balancing flow estimates among neighboring regions. Flow estimates are iteratively balanced across transmission lines common with neighboring regions as well as with respect to the phase angles at their respective boundary buses. We attempt to reduce overall demand violation by determining a local, customized production target $\bar{p}_{r,t}$ based on global demand violation. Computing a customized production target for every region

has been shown to yield smooth convergence in large scale decentralized methods especially from the operational standpoint [50]. In our decentralized formulation represented by Problem (5.3), we assume that operational decision including commitment and production are decoupled for every maintenance epoch. Therefore, for a particular maintenance epoch $m \in M$, let $\boldsymbol{Q}_m^r$ denote the set of inequalities (5.3a)-(5.3h), $\forall t \in T_m$.

$$x_t^g \leq 1 - z_m^g \quad \forall g \in G_r \tag{5.3a}$$

$$P_{min}^g x_t^g \leq y_t^g \leq P_{max}^g x_t^g, \quad \forall g \in G_r \tag{5.3b}$$

$$-\pi_{Dt}^g \leq x_t^g - x_{t-1}^g \leq \pi_{Ut}^g, \ \forall g \in G_r \tag{5.3c}$$

$$-R^g \leq y_t^g - y_{t-1}^g \leq R^g, \ \forall g \in G_r \tag{5.3d}$$

$$\Gamma^{uv}(\theta_t^u - \theta_t^v) = f_t^{uv}, \ \forall u \in \mathcal{U}_r, \forall v \in \mathcal{V}_r^u \tag{5.3e}$$

$$-F_{max}^{uv} \leq \Gamma^{uv}(\theta_t^u - \theta_t^v) \leq F_{max}^{uv}, \ \forall u \in \mathcal{U}_r \cup \mathcal{I}_r, \forall v \in \mathcal{B}_r^u \tag{5.3f}$$

$$\sum_{\forall g \in G_r^u} y_t^g - \delta_t^u + \psi_t^u = \sum_{\forall v \in \mathcal{B}_r^u} [\Gamma^{uv}(\theta_t^u - \theta_t^v)], \forall u \in \mathcal{U}_r \cup \mathcal{I}_r \tag{5.3g}$$

$$\sum_{\forall i \in U_t} \pi_{Ui}^g \leq x_t^g \leq 1 - \sum_{\forall i \in D_t} \pi_{Di}^g, \ \forall g \in G_r, \tag{5.3h}$$

$$U_t = [t - \mu_U^g + 1, t], D_t = [t - \mu_D^g + 1, t]$$

Constraint (5.3a) ensures that a generator that has been placed under maintenance must not have any production. Constraint (5.3b) enforces production limits at each generator while Constraints (5.3c) and (5.3h) enforce minimum up and down-time for each generator. Constraint (5.3d) enforces their respective ramping limitations. Equation (5.3e) establishes the linear relationship between flow and their respective phase angles. Constraint (5.3f) enforces transmission line capacity constraints. Equation (5.3g) balances the demand at each bus with local generation and network flow. Equations (5.3e)-(5.3g) enforce network flow constraints globally. Therefore, the joint CBM and operations problem can be represented

by Problem (5.4).

$$\min \mathcal{L}_r(\Delta) \tag{5.4a}$$

$$\text{s.t.} \sum_{m \in M} z_m^g = 1 \quad \forall g \in G_r \tag{5.4b}$$

$$\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\pi}_U, \boldsymbol{\pi}_D, \boldsymbol{\theta}, \boldsymbol{f} \in \boldsymbol{Q}_m^r, \quad \forall m \in M \tag{5.4c}$$

In Problem (5.4) Constraint (5.4b) represents the maintenance cardinality constraint. Specifically, Constraint (5.4b) ensures that maintenance must be performed on each generator exactly once during the maintenance planning horizon. We estimate two important consensus quantities, i.e. *intermediate flow* denoted by $\bar{F}_t^{uv}$ and *intermediate phase angles* denoted by $\bar{\theta}_t^b$ based on Equations (5.5),(5.6) respectively. These intermediate values are calculated based on the phase angle estimates received from neighboring regions $\forall b \in \mathcal{B}_r$ and $\forall u \in \mathcal{U}_r, \forall v \in \mathcal{V}_r$ for phase angles and flows respectively applied $\forall t \in T$.

$$\bar{\theta}_t^b = \frac{\theta_t^b + \sum\limits_{r' \in \mathcal{N}_r^b} \tilde{\theta}_t^{b,r'}}{|\mathcal{N}_r^b| + 1}, \quad b \in \mathcal{B}^r \tag{5.5}$$

$$\bar{F}_t^{uv} = \frac{\Gamma^{uv}(\tilde{\theta}_t^{u,r'} - \tilde{\theta}_t^{v,r'}) + \Gamma^{uv}(\theta_t^u - \theta_t^v)}{2}, \quad u \in \mathcal{U}_r, v \in \mathcal{V}_r^u, r' \in \mathcal{N}_r^u \tag{5.6}$$

We update the Lagrangian multipliers $\lambda, \phi$ based on Equations (5.7),(5.8) respectively.

$$\lambda_t^b = \lambda_t^b + \rho_\theta(\theta_t^b - \bar{\theta}_t^b) \quad \forall b \in \mathcal{B}_r, \forall t \in T \tag{5.7}$$

$$\phi_t^{uv} = \phi_t^{uv} + \rho_f(F_t^{uv} - \bar{F}_t^{uv}) \quad \forall u \in \mathcal{U}_r, \forall v \in \mathcal{V}_r, \forall t \in T \tag{5.8}$$

We estimate the customized production target for region $r$ based on Equation (5.9).

$$\bar{p}_{r,t} = \sum_{\forall g \in G_r} y_t^g + \frac{\sum_{\forall r \in \mathcal{R}} \varphi_{r,t}}{|\mathcal{R}|} \tag{5.9}$$

In Equation (5.9), $\varphi_{r,t} = \left| \sum_{\forall u \in \mathcal{U}_r \bigcup \mathcal{I}_r} (\delta_t^u - \psi_t^u) - \sum_{\forall g \in G_r} y_t^g \right|$ represents the local violation $\forall t \in T$. We update the associated Lagrangian $\eta$ based on Equation (5.10).

$$\eta_{r,t} = p_{r,t} + \rho_p (p_{r,t} - \bar{p}_{r,t}) \tag{5.10}$$

Therefore, the joint optimization model given by Problem (5.4) describes a Mixed-Integer Quadratic Problem (MIQP) which solves for the maintenance and operations in a decentralized manner. The Lagrangian terms in the model serve as *penalties* for deviating from a position of balance. Convergence occurs when the dualized flow terms become small enough so that the optimization problem given by Problem (5.4) become mathematically equivalent to that of a *centralized* problem as described in [6]. We are now in a position to state Lemma 1.

**Lemma 1.** *Let $\mathcal{L}_r$ represent the objective value of Problem* (5.3) *subject to constraints* (5.4b),(5.4c)*, then*

$$\sum_{m \in M} \underset{\boldsymbol{Q}_m^r}{min}\ \mathcal{L}_r^m(\Delta, \boldsymbol{\alpha}) \leq min\ \mathcal{L}_r(\Delta)$$

*where, $\alpha_g$ represents the dual variable with respect to Constraint* (5.4b)

$$\begin{aligned}\mathcal{L}_r^m(\Delta, \boldsymbol{\alpha}) =& \sum_{t \in T_m} \left[ E_t(\bar{\boldsymbol{\theta}}, \boldsymbol{\phi}) + H_t(\bar{\boldsymbol{F}}, \boldsymbol{\lambda}) + I_t(\bar{\boldsymbol{p}}, \boldsymbol{\eta}) + \nu \sum_{u \in \mathcal{U}_r \cup \mathcal{I}_r} \psi_t^u \right] + \sum_{g \in G_r} \alpha_g \left( \frac{1}{|M|} - z_m^g \right) \\ &+ \sum_{g \in G_r} \left[ \sum_{t \in T_m} \left[ c_t^g x_t^g + d_t^g y_t^g + S_U^g \pi_{Ut}^g + S_D^g \pi_{Dt}^g \right] + (\omega_m^g) z_m^g \right]\end{aligned}$$

*Proof.* Dualizing Equation (5.4b) with the dual variable $\alpha$, we obtain the Lagrangian re-

laxation $\hat{\mathcal{L}}_r$

$$\hat{\mathcal{L}}_r = \mathcal{L}_r(\Delta) + \sum_{g \in G_r} \alpha_g (1 - \sum_{m \in M} z_m^g) = \sum_{m \in M} \mathcal{L}_r^m(\Delta, \boldsymbol{\alpha}) \tag{5.12}$$

Let $\bar{\boldsymbol{\vartheta}} = \{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\pi}_U, \boldsymbol{\pi}_D, \boldsymbol{\theta}, \boldsymbol{f}\}$ be the optimal solution to $\mathcal{L}_r(\Delta)$ (i.e. Problem (5.4)). With this solution fixed, $\bar{\boldsymbol{z}}$ ensures $(1 - \sum_{m \in M} z_m^g) = 0 \quad \forall g \in G_r$, which would enforce i) $\sum_{g \in G_r} \alpha_g (1 - \sum_{m \in M} z_m^g) = 0$, and ii) $\hat{\mathcal{L}}_r|_{\bar{\boldsymbol{\vartheta}}} = \mathcal{L}_r(\Delta)|_{\bar{\boldsymbol{\vartheta}}}$.

Let $\boldsymbol{\vartheta}'$ and $\mathcal{F}$ be the optimal solution, and the set of feasible solutions for $\hat{\mathcal{L}}_r$, respectively. Note that the solution $\bar{\boldsymbol{\vartheta}}$ is still feasible for the lagrangian relaxation (i.e. $\bar{\boldsymbol{\vartheta}} \in \mathcal{F}$, it is trivial to conclude $\hat{\mathcal{L}}_r|_{\boldsymbol{\vartheta}'} = \min_{\vartheta \in \mathcal{F}} \hat{\mathcal{L}}_r|_{\boldsymbol{\vartheta}} \leq \hat{\mathcal{L}}_r|_{\bar{\boldsymbol{\vartheta}}}$. Since, $\hat{\mathcal{L}}_r|_{\boldsymbol{\vartheta}'} \leq \hat{\mathcal{L}}_r|_{\bar{\boldsymbol{\vartheta}}} = \mathcal{L}_r(\Delta)|_{\bar{\boldsymbol{\vartheta}}}$, we conclude:

$$\min_{\bigcup_{m \in M} \boldsymbol{Q}_{\boldsymbol{m}}^r} \hat{\mathcal{L}}_r = \sum_{m \in M} \min_{\boldsymbol{Q}_{\boldsymbol{m}}^r} \mathcal{L}_r^m(\Delta, \boldsymbol{\alpha}) \leq \min \mathcal{L}_r(\Delta) \tag{5.13}$$

We note that minimizing $\hat{\mathcal{L}}_r$ is equivalent to minimizing each of the individual terms $\mathcal{L}_r^m$ since their respective constraint sets $\boldsymbol{Q}_{\boldsymbol{m}}^r$'s are disjoint $\forall m \in M$. □

As a result of Lemma 1, in order to perform one solve of Problem (5.4) it suffices to iteratively solve $\mathcal{L}_r^m$ in parallel $\forall m \in M$ followed by an update of $\alpha_g$ using the subgradient method outlined in [52].

## 5.4 Algorithm Design for Joint Decentralized Maintenance and Operations

We divide the joint, decentralized optimization algorithm into four distinct parts. Each component pertains to the local multithreaded solver, peer-to-peer communication scheme, decentralized optimization frameworks based on convex relaxation and the subgradient methods respectively.

### 5.4.1 Local Multithreaded Optimization Solver

**function** MTOPT($\alpha, \mathcal{L}_r(\Delta), \boldsymbol{Q}^r$)

**for** $m = 1, 2 \ldots |M|$ **using multithreading do**

solve $\min_{Q_m^r} \mathcal{L}_r^m(\Delta, \alpha)$ using Lemma 1

**end for**

**return** $\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{f}, \boldsymbol{\varphi}\}$

**end function**

The entire decentralized joint problem relies on a local optimization solver represented by function `MTOpt`. Owing to its completely decoupled nature along the maintenance planning horizon as detailed in Lemma 1, we apply multithreading to solve Problem (5.3) in parallel to boost computational efficiency. Therefore, given an objective function $\mathcal{L}_r(\Delta)$, a Lagrangian estimate $\alpha$ of Constraint (5.4b) and a constraint set $\boldsymbol{Q}^r$, function `MTOpt` applies $M$ threads to solve one iteration of the local joint problem.

### 5.4.2 Peer to peer Communication

**function** COMMUNICATE($k, \Delta^{k-1}, \boldsymbol{\theta}^k, \boldsymbol{f}^k, \boldsymbol{\varphi}^k$)

send $\boldsymbol{\theta}^{b,k}$ to all regions $r'$, $\forall r' \in \mathcal{N}_r, b \in \mathcal{B}_{r'}$

recieve $\tilde{\boldsymbol{\theta}}^{b,k}$ from all regions $r'$, $\forall r' \in \mathcal{N}_r, b \in \mathcal{B}_{r'}$

send $\boldsymbol{\varphi}^k$ and recieve $\boldsymbol{\varphi}_{r'}^k$ to and from all regions $r' \in \mathcal{R}$

compute $\Delta^k$ based on Equations (5.5)-(5.10)

**if** $||\boldsymbol{f}^k - \bar{\boldsymbol{f}}^k|| < \epsilon$ and $||\bar{\boldsymbol{f}}^k - \bar{\boldsymbol{f}}^{k-1}|| < \epsilon$ **then**

set local convergence to true

if local convergence is true $\forall r \in \mathcal{R}$ then $\Omega \leftarrow 1$

**end if**

**return** $\{\Delta^k, \Omega\}$

**end function**

Communication between neighboring regions occurs according to the steps represented in function `Communicate` which is invoked after every local solve. At every round $k$, estimates of the phase $\boldsymbol{\theta}^k$, flow $\boldsymbol{f}^k$, local violation $\boldsymbol{\varphi}^k$, their corresponding consensus values embodied in $\Delta^{k-1}$ are taken as inputs. The phase angle estimates are sent to neighbors

and are used to compute fresh estimates of the intermediate values $\Delta^k$. The local load violation $\boldsymbol{\varphi}^{r,k}$ is also communicated to all regions and is used to compute the customized local production target. A local convergence check follows based on the primal and dual estimates of the flow. In case of local convergence of all regions, the problem is deemed to have globally converged.

### 5.4.3 Decentralized Optimization with Fixed Maintenance

**function** DECENTFIXEDOPT($\Delta, \boldsymbol{Q}^r$)

$k \leftarrow 0$, $\Delta_0 \leftarrow \Delta$, $\alpha \leftarrow 0$

set global convergence value $\Omega \leftarrow 0$

**while** $\Omega != 0$ **do**

$k \leftarrow k + 1$

$\{\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k, \boldsymbol{\theta}^k, \boldsymbol{f}^k, \boldsymbol{\varphi}^k\} \leftarrow$MTOPT($\alpha, \mathcal{L}_r(\Delta^{k-1}), \boldsymbol{Q}^r$)

$\{\Delta^k, \Omega\} \leftarrow$ COMMUNICATE($k, \Delta^{k-1}, \boldsymbol{\theta}^k, \boldsymbol{f}^k, \boldsymbol{\varphi}^k$)

**end while**

**return** $\{\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k, \Delta^k\}$

**end function**

For solving the joint problem with the maintenance decisions fixed, we follow the sequence of steps represented in function `DecentFixedOpt`. Since maintenance decisions are fixed, the Lagrangian $\alpha$ is also fixed to 0. A multithreaded local solve followed by a peer to peer communication forms one round of the decentralized joint framework. The sequence of computation followed by communication occurs synchronously among all regions until global convergence is achieved.

### 5.4.4 Decentralized Optimization with Local Subgradients

**function** DECENTSGOPT($\Delta, \boldsymbol{Q}^r$)

$k \leftarrow 0$, $\Delta_0 \leftarrow \Delta$, $\alpha_0 \leftarrow 0$

set global convergence value $\Omega \leftarrow 0$

**while** $\Omega != 0$ **do**

$k \leftarrow k+1$, $j \leftarrow 0$, $\Delta^j \leftarrow \Delta^{k-1}$

**while** Constraint (5.4b) not satisfied **do**

$\{\boldsymbol{x}^j, \boldsymbol{y}^j, \boldsymbol{z}^j, \boldsymbol{\theta}^j, \boldsymbol{f}^j, \boldsymbol{\varphi}^j\} \leftarrow \text{MTOPT}(\alpha_j, \mathcal{L}_r(\Delta^j), \boldsymbol{Q}^r)$

$\mathcal{L}^j = \boldsymbol{c}_r^T \boldsymbol{x}^j + \boldsymbol{d}_r^T \boldsymbol{y}^j + \boldsymbol{\omega}_r^T \boldsymbol{z}^j$

$\alpha^{j+1} = \alpha^j + \sigma^j (1 - \sum_{m \in M} z_m^{g,j})$

update $\sigma^{j+1}$ based on Equation (5.14)

$j \leftarrow j+1$

$\{\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k, \boldsymbol{\theta}^k, \boldsymbol{f}^k, \boldsymbol{\varphi}^k\} \leftarrow \{\boldsymbol{x}^j, \boldsymbol{y}^j, \boldsymbol{z}^j, \boldsymbol{\theta}^j, \boldsymbol{f}^j, \boldsymbol{\varphi}^j\}$

**end while**

$\{\Delta^k, \Omega\} \leftarrow \text{COMMUNICATE}(k, \Delta^j, \boldsymbol{\theta}^j, \boldsymbol{f}^j, \boldsymbol{\varphi}^j)$

**end while**

**return** $\{\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k, \Delta^k\}$

**end function**

When the maintenance decisions are released, we utilize the sequence of steps detailed in function `DecentSGOpt`. Specifically, the framework outlined in [52] is utilized to update the Lagrangian value $\alpha$. The step size for the subgradient method is given by Equation (5.14).

$$\sigma_j = \frac{|\mathcal{L}_{UB} - \mathcal{L}^j|}{\sum_{g \in G_r} \left[1 - \sum_{m \in M} z_m^{g,j}\right]^2} \tag{5.14}$$

### 5.4.5 Decentralized Multithreaded Joint Maintenance Algorithm

**Algorithm 5** Multithreaded Decentralized Maintenance and Operations Algorithm

$\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \Delta\}_{FMRC} \leftarrow \text{DECENTFIXEDOPT}(\Delta_0, \boldsymbol{Q}^r_{FMRC})$
$\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \Delta\}_{FMBC} \leftarrow \text{DECENTFIXEDOPT}(\Delta_{FMRC}, \boldsymbol{Q}^r_{FMBC})$
compute $\mathcal{L}_{UB}$ based on $\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\}_{FMBC}$
$\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \Delta\} \leftarrow \text{DECENTSGOPT}(\Delta_{FMBC}, \boldsymbol{Q}^r_{BMBC})$

Table 5.1: Centralized Results

| CGD | Ops Variables | Gross Value (USD $10^4$) | Time (mins) |
|---|---|---|---|
| 4 | 100800 | 714.15 | 120.85 |
| 8 | 201600 | 701.63 | 611.43 |
| 12 | 403200 | 701.86 | 2846.17 |

Based on the four main functions mentioned before, we now present the decentralized joint optimization solution framework in Algorithm 5. We begin by initializing the intermediate values and delineating the three different constraint sets $\boldsymbol{Q}^r_{FMRC}, \boldsymbol{Q}^r_{FMBC}$ and $\boldsymbol{Q}^r_{BMBC}$. Set $\boldsymbol{Q}^r_{FMRC}$ consists of maintenance decisions fixed to the lowest epoch of the maintenance cost function with relaxed commitment variables. As a result, $\boldsymbol{Q}^r_{FMRC}$ is convex in nature. Further, $\boldsymbol{Q}^r_{FMBC}$ represents the constraint set with fixed maintenance and binary commitment variables. It is easy to see that an optimal solution of Problem (5.4) subject to $\boldsymbol{Q}^r_{FMBC}$ is a feasible solution and also an upper bound of the joint problem as well. Moreover, a solution to Problem (5.3) with respect to $\boldsymbol{Q}^r_{FMBC}$ is purely operations oriented. Finally, $\boldsymbol{Q}^r_{BMBC}$ represents a constraint set with released maintenance variables and binary commitment variables.

Initially, Algorithm 5 globally converges with respect to $\boldsymbol{Q}^r_{FMRC}$. Next, the resulting intermediate values are used as input to converge with respect to $\boldsymbol{Q}^r_{FMBC}$. Based on the optimal solution obtained in the previous step, an upper bound of the joint solution is determined that in turn forms the basis of the subgradient method. Warm starting with the help of the intermediate values of the previous step, the local solves based on the subgradient method globally converge to yield the joint maintenance and operations solution.

## 5.5 Experimental Results

Table 5.2: Final Objective Costs (USD $10^4$) of Decentralized Joint CBM and Operations framework

| No. Of | 4 CGD | | | | 8 CGD | | | | 12 CGD | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Regions | Ops | CBM | DC | Gross | Ops | CBM | DC | Gross | Ops | CBM | DC | Gross |
| 50 | 641.16 | 69.82 | 6.85 | 717.82 | 627.89 | 69.80 | 6.96 | 704.65 | 626.12 | 69.86 | 9.96 | 705.95 |
| 60 | 641.05 | 69.82 | 8.79 | 719.66 | 627.03 | 69.83 | 8.96 | 705.82 | 625.70 | 69.80 | 8.43 | 703.94 |
| 75 | 640.32 | 69.84 | 10.48 | 720.65 | 626.82 | 69.87 | 10.72 | 707.40 | 626.78 | 69.81 | 8.34 | 704.93 |
| 80 | 638.32 | 69.83 | 12.37 | 720.52 | 625.13 | 69.81 | 12.24 | 707.18 | 624.93 | 69.83 | 11.27 | 706.03 |
| 90 | 639.25 | 69.84 | 12.52 | 721.61 | 623.68 | 69.82 | 13.17 | 706.67 | 623.64 | 70.01 | 18.82 | 712.47 |
| 100 | 637.93 | 69.85 | 14.34 | 722.12 | 622.80 | 69.91 | 18.05 | 710.76 | 620.61 | 69.88 | 20.07 | 710.56 |



Table 5.3: Computational Performance of Decentralized Joint CBM and Operations framework

| No. Of | No. Of | 4 CGD | | 8 CGD | | 12 CGD | |
|---|---|---|---|---|---|---|---|
| Regions | Threads | Opt Gap(%) | Time (mins) | Opt Gap(%) | Time (mins) | Opt Gap(%) | Time(mins) |
| 50 | 600 | 0.51 | 33.77 | 0.43 | 52.39 | 0.58 | 122.74 |
| 60 | 720 | 0.77 | 44.58 | 0.59 | 53.54 | 0.29 | 170.89 |
| 75 | 900 | 0.90 | 49.8 | 0.82 | 219.17 | 0.43 | 111.50 |
| 80 | 960 | 0.89 | 31.49 | 0.79 | 50.98 | 0.59 | 56.50 |
| 90 | 1080 | 1.04 | 33.24 | 0.71 | 35.02 | 1.51 | 63.06 |
| 100 | 1200 | 1.11 | 41.37 | 1.2 | 39.88 | 1.23 | 368.98 |

For our experiments, we used `Gurobi 7.5` [20] for solving the MIQP problem represented by Problem (5.3) on each node. We evaluate our model on the IEEE 3012 bus case with data derived from `MATPOWER` library [22]. In order to model the regions, we utilize the region decompositions as well as the data provided in [18] for the IEEE 3012 bus case for 150 generators.

### 5.5.1 Computational Architecture

For our experiments, we use an implementation that is based on the Message Passing Interface (MPI)[19] framework for decentralized computation and OpenMP for multithreading. The computational architecture used in our experiments is illustrated on a sample power network divided into three regions depicted in Figure 5.2. The architecture follows a hybrid MPI-OpenMP architecture where each region is assigned one MPI process and every maintenance epoch for the regional subproblem is assigned to one OpenMP thread. Using a distributed memory framework like MPI for decentralization helps us evaluate the algorithm in an environment close to the *real-world*, where each region may represent various utilities. Our multithreaded decentralized computational architecture and software framework can be easily extended to a geographically distributed computational environment.

### 5.5.2 Degradation Modeling

In order to obtain the sensor data necessary to derive the maintenance cost of the assets, we rely on vibration data acquired from a rotating machinery apparatus. Using this experimental setup, condition monitoring is employed to estimate the degradation of the rolling element bearing present in the rotating machine apparatus. We chose roller bearings because they represent an integral component of every rotating machinery including generators and different types of turbines [55, 56]. Vibration signals from the bearings are used to represent generator degradation. We follow an experimental setup outlined in [10] that traces the degradation of bearings from their new state until their failure.

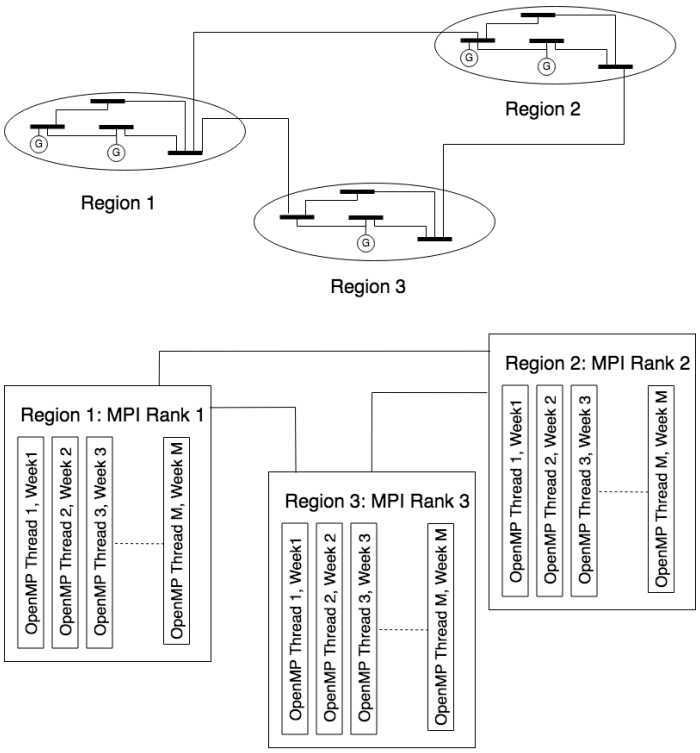



Figure 5.2: Hybrid MPI OpenMP Computational Architecture for solving the Joint CBM and Operations problem

### 5.5.3 Benchmarks and Complexity Measures

In order to benchmark our result, we consider a centralized version of Algorithm 5 with the entire power network considered as a single region. We use the centralized benchmark to rate the performance of the decentralized algorithm by measuring the relative optimality gap. We denote $\gamma$ to be the total optimal objective value comprised of operations and CBM components. $\gamma_{decent}, \gamma_c$ represent the optimal objective for the decentralized joint method and its centralized counterpart respectively. We consider a planning horizon of 3 months comprised of 12 weeks. For highlighting the benefits of our decentralized multithreaded approach with varying problem complexity, we vary the daily operational decision points. Consequently, the operational complexity is quantified by the commitment decisions per generator per day (CGD). Solving a centralized version of the joint problem could be significantly challenging in itself. In our experiments, the centralized version did not converge within 6 hours for even the simplest operational complexity of 4 CGD. Therefore,

our centralized benchmark formulation considers its relaxation without transmission line constraints. It follows that the true centralized optimal objective would be higher than the relaxation, indicating that the reported optimality gap in our results could actually be lower.

### 5.5.4 Computational Results and Observations

Table 5.1 depicts the results of the centralized algorithm with varying number of CGD values. We use the centralized results for benchmarking our decentralized algorithm. Table 5.2 depicts the operational (Ops), CBM, Demand Curtailment (DC) and Gross Total cost with respect to increasing CGD values with varying number of regions. Further, Table 5.3 shows the computational performance of the decentralized algorithm with respect to the optimality gap, the computational time incurred as well as the total number of threads used. Specifically, we show the results with varying CGD and region cases. We observe that with increasing complexity of the problem as indicated by higher CGD values, the computational time keeps increasing. Further, Table 5.2 shows that our decentralized approach yields highly stable solutions with respect to optimal maintenance decisions as well as operational decisions. On the other hand Table 5.3 yields numerous interesting insights concerning the computational performance of the decentralized algorithm. First, we observe that with increasing number of regions, the time incurred by our algorithm does not increase by a significant amount. In fact, our approach is seen to be highly stable with increasing number of regions thereby highlighting scalability. Moreover, we can also note that with increasing complexity of the problem denoted by the CGD value, the time incurred increases approximately in a linear fashion thereby demonstrating computational efficiency. Lastly, we observe that in terms of quality of solution as indicated by the optimality gap, our algorithm retains its stability with increasing number of regions across varying CGD values.

## 5.6 Conclusion and Future Work

In this chapter we develop a decentralized, multithreaded framework for the joint CBM and operational problem designed for large scale power systems. Our solution involves, decomposing a given power network topology into multiple regions and using ADMM to formulate a joint optimization model. Such a formulation based on decentralization, allows preservation regional data privacy. We solve the optimization model in a decentralized manner wherein each region holds its own local subproblem and cooperates with its neighbors by exchanging flow estimates. We further leverage multithreading at every regional subproblem to bolster the computational efficiency of our solution. We demonstrate the convergence of our algorithm based on experiments on the large scale IEEE 3012 bus case incorporating varying degrees of region decompositions. The results demonstrate that our decentralized algorithm can provide good solution quality, scalability and efficiency with full privacy of sensor data while being robust to varying problem complexities. As part of our future work, we intend to explore computational efficiency stemming from an asynchronous model of operation as well as integrating differential privacy for consensus.

# CHAPTER 6
# DECENTRALIZED AND SECURE GENERATION MAINTENANCE WITH DIFFERENTIAL PRIVACY

## 6.1 Introduction

Planning problems are the cornerstone for efficient functioning of transmission systems in large scale power systems. Some examples of critical planning problems include economic dispatch [57], optimal power flow[58], unit commitment (UC) [9] and maintenance [45, 46]. Optimal planning decisions are subject to operational and reliability constraints [47] which ultimately require solving large scale optimization problems. Recently, a growing body of literature has focused on the use of consensus driven, decentralized optimization strategies to address issues of computational scalability and data localization [59, 18, 39, 50, 51] pertaining to the power systems planning problem. Despite their success, decentralized methods require disclosure of network flow estimates to their peers in order to compute optimal decisions. Such disclosures typically take place over public facing communication channels such as the internet [50] leading to privacy risks emanating from a malicious third party. To address this risk, we develop a novel decentralized optimization framework that leverages differential privacy [5], for protecting network flow estimates.

Differential privacy is a widely used method to protect the privacy of datasets intended to be communicated through public domains [5, 60]. Differential privacy driven approaches involve injecting a randomized noise in order to obfuscate the real underlying data record. The injected randomized noise can be designed so as to facilitate theoretical guarantees bounding the loss of privacy [5]. Differential privacy thereby ensures that the probability of extracting the real value from a noisy dataset by any external entity remains remarkably low. As a result, differential privacy forms an attractive option to preserve privacy of network

flow values in decentralized planning problems.

In order to demonstrate our framework, we develop a decentralized formulation of the generation maintenance problem [61] whose solution is critical to the scheduling of operations and maintenance over a designated planning window. Being a fundamental problem in power systems, generation maintenance is particularly susceptible to privacy and scalability issues. There are a number of unique aspects of the generation maintenance problem that makes it an interesting problem to study in our setting. First, it consists of binary decisions for generator maintenance across discrete time windows as well as hourly binary commitment decisions. Owing to binary decisions as well as a planning horizon of a week, the generation maintenance problem is large scale and mixed integer in nature. As widely-documented in decentralized optimization literature [39], mixed integer variables introduce significant challenges in model coordination. Second, we are focusing on sensor-driven generation maintenance which harnesses highly-sensitive asset-health data from generation assets. This information, if compromised, can lead to significant risks in asset safety and operational vulnerabilities. Third, the generation maintenance problem consists of multiple interdependent UC problems augmented with maintenance variables, making it a significantly more challenging problem than UC. It is evident, therefore, that the framework developed in this chapter can be directly applied to the simpler, decentralized UC formulation as well.

Our decentralized formulation decomposes the power network topology (i.e. spatial decomposition) into several regions which may represent various utility stakeholders or regional monitoring centers. To decompose the problem, we first relax the network flow constraints pertaining to transmission lines connecting two regions yielding regionally independent local subproblems. These constraints are dualized and incorporated into the objective function of these local subproblems to ensure coordination across subproblems. More specifically, the network flow estimates corresponding to the dualized constraints are balanced between neighboring regions through the iterative application of the Alternating

Direction Method of Multipliers (ADMM) [12]. ADMM is a key component of decentralized operational planning strategies [18, 50]. In our framework, the ADMM methods will communicate a differentially private version of the phase angles across regions, from which the corresponding flow values will be estimated.

Our strategy for differential privacy is based on the numerous benefits stemming from the relationship between the phase angles and flow. First, owing to their linear relation, a noise injection on the phase angles leads to a corresponding linear transformation being injected to the flow as well. Second, we note that in decentralized formulations of the planning problems, phase angles are primarily meant for computing the flow [39, 50]. Therefore, we can choose a noise to be injected on the phase angles such that its linear transformation leads to differential privacy guarantees on the corresponding flow values. If chosen carefully, the noise could also ensure privacy of flow values estimated from phase angle estimates emanating from different iterations and/or regions as well. Such a feature is significantly useful in the asymptotic sense, when the true phase angle estimates across multiple iterations and regions are very close to each other.

Further, we also note that improved convergence is all the more important in a differentially private setting for a dynamically evolving process (i.e. coordination mechanism causes the underlying phase angle and flow estimates to change through iterations). To discover this dynamic underlying convergence, we adopt an Exponentially Weighted Moving Average (EWMA) that processes the noisy phase angle estimates leading to faster convergence. In order to balance the trade-off between faster convergence and better solution quality, we employ the use of a regional control chart based on the Central Limit Theorem (CLT). Our control chart is applied on the consensus quantities estimated at every iteration on each region and is geared towards bringing an out of control process to in control. As a result, the control chart mechanism stabilizes the solution quality with respect to varying noise levels while retaining good convergence behavior.

Our contributions in this chapter can be summarized as follows:

- We develop an ADMM based differentially private, decentralized planning framework for the generation maintenance problem. The mixed integer nature of the problem renders the ADMM application a significant challenge even without differential privacy. In our setting, this challenge is compounded by the use of differential privacy.

- We propose well-suited noise injection strategies that leverage the structure of the problem. Our approach injects an engineered noise at the level of the phase angles, that culminates in differential privacy of flow values between regions.

- We develop an EWMA-based mechanism to improve convergence at the presence of dynamically changing flow estimates. We evaluate the EWMA outputs within a CLT based control chart for stabilizing the solution quality.

- We provide a High Performance Computing (HPC) driven implementation for simulating our framework under a diverse set of scenarios.

Our experiments on the 8 and 12 region decompositions of the 118 bus case demonstrates that our approach is robust to a wide variety of noise scenarios and convergence limits. Extensive experiments demonstrate that the proposed approach provides stable solution quality that rivals its benchmark without any differential privacy.

## 6.2 Related Works

In transmission system planning problems, the operational and reliability constraints rely on infrastructure data that is held locally by the various utility stakeholders [39]. In order to solve planning problems, infrastructure data must be aggregated at a centralized location leading to privacy and cyber security risks [50, 18]. In addition to revealing private and sensitive infrastructure data of the stakeholders, such a centralized computational model also leads to communication bottlenecks on the central location [62]. In the context of mixed integer power system planning problems, decentralized unit commitment was first

proposed in [18] as a means for obtaining optimal UC decisions for networks without central control. An asynchronous version of the decentralized UC framework was proposed in [39] with the purpose of improving computational efficiency. An extended version of the asynchronous decentralized model was discussed in [50], which provided improved solution quality for a large scale problem setting. More recently, the work done in [51] proposes a decentralized UC formulation using the Power Transfer Distribution Factor (PTDF) as a means to improve scalability.

In this chapter, we study generation maintenance problem that jointly identifies optimal maintenance and UC decisions. UC problems studied in [18, 39, 50, 51] form a subproblem within our setting. In generation maintenance, there is rich literature in coordination mechanisms between generation companies and market operators in a deregulated market setting [45, 46, 63]. Our focus is on integrated operations and maintenance problems that solve for optimal maintenance as well as operational schedules subject to network constraints [47, 48, 49]. Generation maintenance problems typically use periodic maintenance policies, which require fixed time-based requirements for generation assets based on manufacturer recommendations and field experience (i.e. yearly major overhaul requirements). In contrast, our approach uses sensor-data to conduct condition-based maintenance strategies as proposed in [3, 6]. We integrate asset failure risks obtained through sensor data, within a joint optimization of operations and maintenance decisions. We study the short term periodic maintenance problem setting as in [61]. Our approach could potentially be adapted to other maintenance problems as well as operational paradigms. In our setting, any compromise to information security can reveal network-wide vulnerabilities that can lead to cyber-physical attacks on power systems, and opportunities for market manipulation [64, 65].

Most differential privacy approaches in power systems are geared towards public release of operational power flow (OPF) data for benchmarking purposes. They address issues such as: quantifying the dynamics between injected noise and topology [66]; inject-

ing noise into the OPF constraint set while guaranteeing solution accuracy [67], perturbing transmission line parameters [68] and hiding sensitive load locations as well as values [69, 70]. Lastly, the authors in [71] adopt a distributed, differentially private, ADMM driven approach to solve the AC-OPF problem by perturbing the demand at each bus.

In contrast to the above works, our framework is meant for utility stakeholders to schedule their local operations and maintenance subject to global consensus over network constraints. Incorporating differential privacy in decentralized formulations of mixed integer power system planning problems largely remains understated, and has not been studied in a generation maintenance setting. Due to unique challenges in generation maintenance (e.g. large scale and mixed integer nature of the problem, and dynamically changing phase angle values), existing differential privacy approaches do not scale to our problem setting, requiring us to develop novel approaches to address these challenges.

## 6.3 Decentralized Short Term Maintenance

Our differential privacy driven technique is motivated by the recent developments of decentralized computational methods in power systems. As a result, our privacy preserving problem formulation comprises three main components which are detailed in this section. First, we discuss a decentralized formulation that employs mixed integer optimization techniques to yield maintenance and operational decisions including hourly commitment schedules. Second, we discuss our novel differential privacy driven information exchange that is utilized for obtaining the ADMM balance of flow. Lastly, we present our privacy preserving optimization framework that incorporates EWMA as well as control charts for stable convergence and superior solution quality.

### 6.3.1 Decentralized Short Term Maintenance and Commitment

We propose a decentralized formulation based on regional decomposition leading to the respective regional subproblems. From a practical standpoint, each region may denote a

subsidiary of the utility company in a vertically integrated market or a utility company in a deregulated market. Therefore, every region is comprised of local generators and buses subject to its own operational constraints.

We show the regional decomposition of a sample network with the help of Figure 6.1. The notations are elaborated in Appendix C. Our example network consists of 3 regions

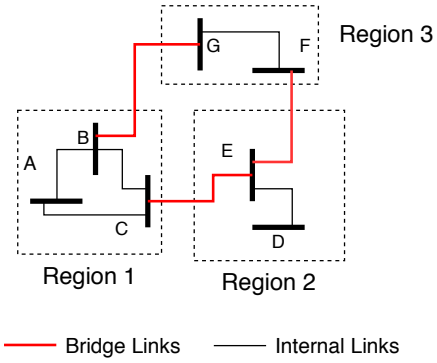



Figure 6.1: Partition of Network topology into regions.

with boundary and foreign bus categorization for each region defined as follows:

- Region 1: $\mathcal{U}_1 = \{B, C\}$, $\mathcal{V}_1 = \{G, E\}$
- Region 2: $\mathcal{U}_2 = \{E\}$, $\mathcal{V}_2 = \{F, C\}$
- Region 3: $\mathcal{U}_3 = \{G, F\}$, $\mathcal{V}_3 = \{B, E\}$

The regional subproblem seeks to minimize the objective cost as represented by Problem (6.1) as follows. For simplicity we consider the vector form of the variables as necessary.

$$\begin{aligned}\min_{\boldsymbol{z},\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\phi}} \quad & \mathcal{L}_r(\bar{\boldsymbol{\theta}}_k, \bar{\boldsymbol{F}}_k, \boldsymbol{\lambda}_k, \boldsymbol{\phi}_k) = \sum_{t\in T}\sum_{g\in G_r} D_g y_t^g + C_g x_t^g + \sum_{m\in M}\sum_{g\in G_r^d} K_g^m z_g^m \\ & + \sum_{t\in T}\sum_{b\in\mathcal{B}_r} [\lambda_t^b|\theta_t^b - \bar{\theta}_t^b| + \frac{\rho_\theta}{2}(\theta_t^b - \bar{\theta}_t^b)^2] \\ & + \sum_{t\in T}\sum_{u\in\mathcal{U}_r}\sum_{v\in\mathcal{V}_r^u} [\phi_t^{uv}|f_t^{uv} - \bar{f}_t^{uv}| + \frac{\rho_f}{2}(f_t^{uv} - \bar{f}_t^{uv})^2]\end{aligned} \tag{6.1}$$

The regional objective function represented by Problem (6.1) consists of a dispatch cost

component (the term with $D_g y_t^g$) a commitment cost component (the term with $C_g x_t^g$) and a dynamic maintenance cost component (the term with $K_g^m z_g^m$) that is stored locally. The dynamic maintenance cost $K_g^m$ is dynamically evaluated based on sensor-driven predictions on the remaining life of generator $g$. For more information on this cost factor, we refer the reader to [3, 6]. In addition, the objective function also includes ADMM penalty terms imposed to balance flow estimates among neighboring regions. Flow estimates are iteratively balanced across transmission lines between neighboring regions through an iterative process. After every local solve of Problem (6.1), the fresh estimates of phase angles are shared with neighbors in order to balance flows. Based on estimates received from neighbors, a consensus quantity can be estimated for flow as well as phase angles denoted by $\bar{f}, \bar{\theta}$ respectively.

Commitment, production and maintenance decisions are computed based on locally held constraints. Each maintenance window is comprised of several operational decision points such that,

$$T = \bigcup_{m=1}^{|M|} T_m, \text{ where, } T_m = \left\{ t | t \in \left[ \frac{m|T|}{|M|} \dots \frac{(m+1)|T|}{|M|} \right] \right\}$$

The regional subproblem for the joint operations and maintenance optimization is subject to a number of local constraints for $t \in T_m$, $m \in M$ represented by the set $Q^r$:

$$x_t^g \leq 1 - z_m^g \quad \forall g \in G_r \tag{6.2a}$$

$$P_{min}^g x_t^g \leq y_t^g \leq P_{max}^g x_t^g, \quad \forall g \in G_r \tag{6.2b}$$

$$-\pi_{Dt}^g \leq x_t^g - x_{t-1}^g \leq \pi_{Ut}^g, \ \forall g \in G_r \tag{6.2c}$$

$$-R^g \leq y_t^g - y_{t-1}^g \leq R^g, \ \forall g \in G_r \tag{6.2d}$$

$$\Gamma(uv)(\theta_t^u - \theta_t^v) = f_t^{uv}, \ \forall u \in \mathcal{U}_r, \forall v \in \mathcal{V}_r^u \tag{6.2e}$$

$$-F_{max}^{uv} \leq \Gamma^{uv}(\theta_t^u - \theta_t^v) \leq F_{max}^{uv}, \ \forall u \in \mathcal{U}_r \cup \mathcal{I}_r, \forall v \in \mathcal{B}_r^u \tag{6.2f}$$

$$\sum_{\forall g \in G_r^u} y_t^g - \delta_t^u + \psi_t^u = \sum_{\forall v \in \mathcal{B}_r^u} [\Gamma^{uv}(\theta_t^u - \theta_t^v)], \forall u \in \mathcal{U}_r \cup \mathcal{I}_r \tag{6.2g}$$

$$\sum_{\forall i \in U_t} \pi_{Ui}^g \leq x_t^g \leq 1 - \sum_{\forall i \in D_t} \pi_{Di}^g, \ \forall g \in G_r, U_t = [t - \mu_U^g + 1, t], D_t = [t - \mu_D^g + 1, t] \tag{6.2h}$$

Constraint (6.2a) ensures that generators placed under maintenance do not have any production, where $M(t)$ represents the maintenance window corresponding to the operational decision point $t$. Constraint (6.2b) enforces limits on each generator's maximum and minimum production levels. Constraints (6.2c) and (6.2h) enforce minimum up and down-time for each generator. Constraint (6.2d) enforces generator ramping limits. Equation (6.2e) enforces the linear relationship between flows and their respective phase angles. Constraint (6.2f) limits the transmission line capacity. Equation (6.2g) balances demand at each bus with corresponding generation and network flow. Equations (6.2e)-(6.2g) enforce network flow constraints globally.

In addition to these constraints, we also enforce that every degraded generator $g \in G_r^d$ in region $r$ is maintained within the planning horizon:

$$\sum_{m \in \mathcal{M}} z_m^g = 1 \quad \forall g \in G_r^d \tag{6.3}$$

## 6.4 Differential Privacy For Decentralized Planning

Decentralized optimization frameworks for power system planning problems rely on ADMM based, iterative, flow and phase angle balancing between neighbor regions [39, 50, 51]. In order to converge to the global optimum, regions share phase angle estimates with neighbors which can in turn also be used to estimate flow using Equation (6.2e) [39, 50]. Based on the phase angle and flow estimates received, regions compute consensus quantities $\bar{\theta}, \bar{f}$ as shown in Problem (6.1). These consensus quantities are critical for updating the Lagrangian duals $\lambda, \phi$ and therefore strongly influence global convergence to the optimal so-

lution. In this chapter, we focus on computing globally optimal, short term maintenance and operational decisions for every region in a decentralized fashion while employing differential privacy for protecting the flow.

### 6.4.1 Differential Privacy Primer

Before delving into a detailed discussion of our framework, we briefly review key theorems pertaining to differential privacy that are vital to our framework.

**Theorem 1.** ***Definition*** *[5] : A randomized mechanism $\mathcal{M}$ with domain $\mathcal{R}$ and sensitivity $\omega > 0$ is said to preserve $\epsilon$-differential privacy for some $\epsilon \geq 0$ $\forall x, x' \in \mathcal{R}$ with $||x - x'||_1 \leq \omega$, if the following relation holds:*

$$Pr(\mathcal{M}(x) \in \mathcal{R}) \leq e^{\epsilon} Pr(\mathcal{M}(x') \in \mathcal{R})$$

In other words, Theorem 1 ensures that the probability of computing the exact distance of the true value $x$ from its perturbation $\mathcal{M}$ is low. The parameter $\epsilon$ represents a privacy budget, with a smaller value favoring a higher degree of privacy. The sensitivity parameter $\omega$ ensures obfuscation of values close to each other while maintaining the relative difference of values far apart.

**Theorem 2.** ***Post Processing Immunity*** *[5] : Given a mechanism $\mathcal{M}$ that preserves $\epsilon$-differential privacy, then for any function $g$, the functional composition $g \circ \mathcal{M}$ also preserves $\epsilon$-differential privacy.*

An important result of differential privacy pertains to post processing immunity encapsulated in Theorem 2. The post processing immunity implied by Theorem 2 means that once differential privacy has been applied on any element of the domain $\mathcal{R}$, no further privacy can be lost with application of any arbitrary function by a third party.

**Theorem 3.** ***Adaptive Composition*** *[60] : Given mechanisms $\mathcal{M}_1, \mathcal{M}_2$, which ensure $\epsilon_1, \epsilon_2$ differential privacy respectively, a mechanism $\mathcal{M}(x) = \mathcal{M}_2(x, \mathcal{M}_1(x))$ also preserves differential privacy.*

Theorem 3 provides a critical result that is especially useful for developing a decentralized, iterative optimization framework, wherein existing results depend on consensus estimates obtained from the previous iteration.

Lastly, in Theorem 4, we describe the Laplacian mechanism, which is one of the most commonly used techniques for ensuring differential privacy.

**Theorem 4.** ***Laplacian mechanism*** *[5] : Given a function $g$ with domain $\mathcal{R}$ and $\omega > 0$, a mechanism $\mathcal{M}(x) = g(x) + w$, where $x \in \mathcal{R}$ and $w \sim Lap(0, \frac{\omega}{\epsilon})$ is $\epsilon$-differentially private.*

Theorem 4 guarantees that the mechanism $\mathcal{M}$ based on the Laplacian distribution with zero mean and standard deviation $\frac{\omega}{\epsilon}$ preserves $\epsilon$-differential privacy.

### 6.4.2 Differential Privacy for Flow

In order to protect regional flow values with differential privacy, we exploit the linear relationship between phase angles and the corresponding flow variables. As a result, we propose Theorem 5 which relies on an important property of Laplace distributions stated in Lemma 2 [72].

**Lemma 2.** *Given two random variables $X, Y \sim Exp(\omega)$, the random variable $X - Y \sim Lap(0, \omega)$.*

**Theorem 5.** *Given a transmission line across $b_1 b_2$ between regions $r_1, r_2$, such that $b_1 \in \mathcal{U}_{r_1}, b_2 \in \mathcal{V}_{r_1}^{b_1}, b_2 \in \mathcal{U}_{r_2}, b_1 \in \mathcal{V}_{r_2}^{b_2}$, a mechanism $\mathcal{T}$ given by*

$$\mathcal{T}(\theta_t^{b_i}) = \theta_t^{b_i} + \alpha_t^{b_i}, \text{ where, } \alpha_t^{b_i} \sim Exp\Big(\frac{\omega}{|\Gamma(b_1 b_2)|\epsilon}\Big), \; i \in 1, 2$$

*preserves $\epsilon$-differential privacy of flow $f_t^{b_1 b_2}$ defined as $\mathcal{M}'(\theta_t^{b_1}, \theta_t^{b_2}) = \Gamma(b_1 b_2)\Big(\mathcal{T}(\theta_t^{b_1}) - \mathcal{T}(\theta_t^{b_2})\Big)$.*

*Proof.* Consider a Laplacian distribution $Lap(0, \tilde{\omega}/\epsilon)$, where $\tilde{\omega} = \omega(\Gamma(b_1 b_2))^{-1}$. Substituting $\mathcal{T}(\theta_t^u)$ in Equation (6.2e), we have

$$\mathcal{M}'(\theta_t^{b_1}, \theta_t^{b_2}) = \Gamma(b_1 b_2)\Big(\mathcal{T}(\theta_t^{b_1}) - \mathcal{T}(\theta_t^{b_2})\Big) \tag{6.4}$$

$$= \Gamma(b_1 b_2)(\theta_t^{b_1} - \theta_t^{b_2}) + \psi_t^{b_1, b_2} \tag{6.5}$$

where (6.4), $\psi_t^{b_1,b_2} = \Gamma(b_1 b_2)(\alpha_t^{b_1} - \alpha_t^{b_2})$. Based on Lemma 2, we can claim that the random variable given by $\frac{\psi_t^{b_1,b_2}}{\Gamma(b_1 b_2)} = (\alpha_t^{b_1} - \alpha_t^{b_2})$ follows $Lap(0, \tilde{\omega}/\epsilon)$. Owing to the Laplace distribution being symmetric, we can state that the random variable $\psi_t^{b_1,b_2} \sim Lap(0, \omega/\epsilon)$. Since $\Gamma(b_1 b_2)(\theta_t^{b_1} - \theta_t^{b_2}) = f_t^{b_1,b_2}$ denotes the real flow and $\psi_t^{b_1,b_2} \sim Lap(0, \omega/\epsilon)$ is Laplace distributed, Theorem 4 concludes that mechanism $\mathcal{M}'$ preserves the $\epsilon$-differential privacy of flow. □

An important consequence of Theorem 5 is that the phase angle values with an exponential perturbation directly lead to imposing the $\epsilon$-differential privacy for the flow. Therefore, our iterative scheme is based on sharing the phase angles and by extension the differentially private flow values as well.

**Corollary 5.1.** *Given $\theta_t^{b_1,k,r_1}, \alpha_t^{b_1,k,r_1}$ for bus $b_1$ at iteration $k$ and region $r_1$, and $\theta_t^{b_2,j,r_2}, \alpha_t^{b_2,j,r_2}$ for bus $b_2$ at iteration $j$ and region $r_2$, respectively; the mechanism $\mathcal{M}'(\theta_t^{b_1,k,r_1}, \theta_t^{b_2,j,r_2})$ still preserves the $\epsilon$-differential privacy of the corresponding flow.*

*Proof.* We note that

$$\mathcal{T}(\theta_t^{b_1,k,r_1}) - \mathcal{T}(\theta_t^{b_2,j,r_2}) = (\theta_t^{b_1,k,r_1} - \theta_t^{b_2,j,r_2}) + (\alpha_t^{b_1,k,r_1} - \alpha_t^{b_2,j,r_2}) \tag{6.6}$$

Following a similar reasoning as presented in Theorem 5, Lemma 2 indicates that the noise given by $\alpha_t^{b_1,k,r_1} - \alpha_t^{b_2,j,r_2}$ follows a Laplacian distribution ensuring $\epsilon$-differential privacy.

From Theorem 2, it also follows that $\Gamma(b_1 b_2)(\mathcal{T}(\theta_t^{b_1,k,r_1}) - \mathcal{T}(\theta_t^{b_2,j,r_2}))$ also remains differentially private. □

Corollary 5.1 establishes the fact that phase angle values derived from a combination of historically observed values cannot be used to infer actual flow values. Thus, Corollary 5.1 is especially useful in an asymptotic sense when the real phase angle estimates across multiple iterations and regions might be close. Finally, we note that the indices $1$ and $2$ were used for ease of exposition, and the theorem applies for flow estimates of any transmission line.

## 6.5 Algorithm Design for Decentralized Differential Privacy

We employ Theorem 5 in order to construct our decentralized algorithm with differential privacy. Our algorithm consists of two key components pertaining to improved convergence and added stability. For improving convergence we use an EWMA based consensus technique to balance phase angles and flow. On the other hand, for added stability, we propose the use of CLT control charts.

### 6.5.1 EWMA based Consensus Mechanism for ADMM

At iteration $k$, each region shares the noisy phase angles $\hat{\theta}_t^{b,k} = \theta_t^{b,k} + 2\alpha_t^{b,k}$, $\forall b \in \mathcal{U}_r \bigcup \mathcal{V}_r$ with its neighbors. The noisy phase angle estimates are utilized to compute the noisy flow estimates $\hat{f}_t^{uv,k}$, $\forall u \in \mathcal{U}_r, \forall v \in \mathcal{V}_r^u$ based on Theorem 5.

Based on the received values from neighbor $r' \in \mathcal{N}_r$, we estimate the two consensus terms for transmission line $uv$ separately leading to the *intermediate flow*,*intermediate phase angle* denoted by $\bar{f}_t^{uv,k}$, $\bar{\theta}_t^{b,k}$ respectively. However, for added stability we apply an Exponentially Weighted Moving Average (EWMA) to the received values from the neighbor. EWMA leads us to Equations (6.7) to (6.12) with the mixing factor denoted by $\eta$. Specifically, for phase angles of bus $b \in \{u, v\}$ corresponding to transmission line $uv$ we

have,

$$\tilde{\theta}_t^{b,0,r'} = \hat{\theta}_t^{b,0,r'} \tag{6.7}$$

$$\tilde{\theta}_t^{b,k,r'} = \eta(\hat{\theta}_t^{b,k,r'}) + (1-\eta)(\tilde{\theta}_t^{b,k-1,r'}) \tag{6.8}$$

$$\bar{\theta}_t^{b,k} = \frac{\theta_t^{b,k} + \tilde{\theta}_t^{b,k,r'}}{2} \tag{6.9}$$

Similarly for flow on transmission line $uv, \forall u \in \mathcal{U}_r, \forall v \in \mathcal{V}_r^u$ we have,

$$\tilde{f}_t^{uv,0,r'} = \hat{f}_t^{uv,0,r'} \tag{6.10}$$

$$\tilde{f}_t^{uv,k,r'} = \eta(\hat{f}_t^{uv,k,r'}) + (1-\eta)(\tilde{f}_t^{uv,k-1,r'}) \tag{6.11}$$

$$\bar{f}_t^{uv,k} = \frac{f_t^{uv,k} + \tilde{f}_t^{uv,k,r'}}{2} \tag{6.12}$$

We update the Lagrangian multipliers as follows

$$\lambda_t^{b,k} = \lambda_t^{b,k-1} + \rho_\theta(\theta_t^{b,k} - \bar{\theta}_t^{b,k}) \quad \forall b \in \mathcal{B}_r, \forall t \in T \tag{6.13}$$

$$\phi_t^{uv,k} = \phi_t^{uv,k} + \rho_f(f_t^{uv,k} - \bar{f}_t^{uv,k}) \quad \forall u \in \mathcal{U}_r, \forall v \in \mathcal{V}_r, \forall t \in T \tag{6.14}$$

The optimization model given by Problem (6.2) describes a Mixed-Integer Quadratic Problem (MIQP) which solves for the maintenance and operations in a decentralized manner. Since we have the presence of binary variables in $z$, our problem is *non-convex*. As a result, it becomes much harder than traditional convex schemes to achieve convergence in a decentralized manner. Recent works have demonstrated the successful application of ADMM for solving decentralized non-convex problems. The maintenance cost is fed as input to the data and is derived from the work done in [6].

The Lagrangian terms in the model serve as *penalties* for deviating from a position of balance. Convergence occurs when these terms become small enough such that the

optimization problem given by Problem (6.2) becomes mathematically equivalent to that of a *centralized* problem as described in [6]

### 6.5.2 CLT Control Chart based Convergence Criteria

In order to bolster the robustness of overall solution quality against the noise, we employ a CLT based control chart as our convergence criteria. Our use of the CLT driven control chart is driven by the symmetric nature of the Laplacian distribution. Consider $\Theta$ defined as,

$$\Theta = \sum_{k=1}^{S_w} (\theta_t^{u,k,r} + \alpha_t^{u,k,r} - \theta_t^{u,k,r'} - \alpha_t^{u,k,r'}) \tag{6.15}$$

where $S_w$ are the number of iterations which form one point on the CLT. Asymptotically, we expect the real phase angles from different regions obtained at iteration $k$ to match. Applying CLT on their corresponding noise leads us to require that $\Theta \sim N(0, \frac{2\tilde{\omega}^2}{S_w})$. Under CLT, the phase angle control process triggers an alarm whenever $\Theta > \sqrt{\frac{2\tilde{\omega}^2}{S_w}}$ or $\Theta < -\sqrt{\frac{2\tilde{\omega}^2}{S_w}}$. A single value of $\Theta$ forms one point on the control chart for the respective bus. We assume that our stopping criteria triggers local convergence only when $S_p$ points on the control chart lead to a single alarm. This implies that local convergence can only be certified in multiples of $S_w \cdot S_p$ iterations.

### 6.5.3 Decentralized Maintenance and Operations Algorithm

Our decentralized and differentially private algorithm relies on two key components pertaining to the local optimization solves on every region and the peer-to-peer communication scheme. For notational simplicity, we denote $\Delta = \{\bar{\boldsymbol{\theta}}_k, \bar{\boldsymbol{F}}_k, \boldsymbol{\lambda}_k, \boldsymbol{\phi}_k\}$.

*Local Optimizer*

The subroutine `OptSolve` represents the local optimization that occurs at every region. Specifically, our regional solver consumes the constraint set represented by $Q$ in addi-

tion to the objective function (6.1) derived from latest consensus variables as well as their corresponding multipliers. It returns the latest estimates pertaining to the commitment, production, maintenance decisions as well as unperturbed phase angles and flow estimates.

**function** OPTSOLVE($\mathcal{L}_r(\Delta), \boldsymbol{Q}$)

$\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{f}\} \leftarrow \min \mathcal{L}_r(\Delta)$ subject to $Q$

**return** $\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{f}\}$

**end function**

*Differential Privacy driven Communication scheme*

The subroutine `DPCommunicate` represents the peer-to-peer to communication scheme that adopts the phase angle and flow based differential privacy scheme given in Theorem 5. Exponentially perturbed phase angle estimates are shared with neighbors and consensus quantities are computed at every iteration. Local convergence occurs when the phase angle residual values are below the primal and dual tolerance ($\beta_p, \beta_d$) respectively and the number of alarms $\kappa$ in the region is less than $|\mathcal{U}_r \cup \mathcal{V}_r|$.

**function** DPCOMMUNICATE($k, \Delta^{k-1}, \boldsymbol{\theta}^k, \boldsymbol{f}^k$)

send $\mathcal{M}(\boldsymbol{\theta}^{b,k})$ to all regions $r'$, $\forall r' \in \mathcal{N}_r, b \in \mathcal{B}_{r'}$

receive $\tilde{\boldsymbol{\theta}}^{b,k}$ from all regions $r'$, $\forall r' \in \mathcal{N}_r, b \in \mathcal{B}_{r'}$

compute $\bar{\boldsymbol{\theta}}_k, \bar{\boldsymbol{f}}_k, \boldsymbol{\lambda}_k, \boldsymbol{\phi}_k$ based on Equations (6.9)-(6.14)

**if** $||\boldsymbol{\theta}^k - \bar{\boldsymbol{\theta}}^k|| < \beta$ & $||\bar{\boldsymbol{\theta}}^k - \bar{\boldsymbol{\theta}}^{k-1}|| < \beta$ & $\kappa < |\mathcal{U}_r \cup \mathcal{V}_r|$ **then**

set local convergence to true

if local convergence is true $\forall r \in \mathcal{R}$ then $\Omega \leftarrow 1$

**end if**

**return** $\{\Delta^k, \Omega\}$

**end function**

*Regional Solver*

The subroutine `DecentDPOpt` represents the regional solver which iteratively invokes the local optimizer followed by a round of differentially private message exchange with the neighbors.

**function** DECENTDPOPT($\Delta, \boldsymbol{Q}^r$)

$k \leftarrow 0$, $\Delta_0 \leftarrow \Delta$,

set global convergence value $\Omega \leftarrow 0$

**while** $\Omega \neq 0$ **do**

$k \leftarrow k + 1$

$\{\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k, \boldsymbol{\theta}^k, \boldsymbol{f}^k\} \leftarrow$ OPTSOLVE($\mathcal{L}_r(\Delta^{k-1}), \boldsymbol{Q}^r$)

$\{\Delta^k, \Omega\} \leftarrow$ COMMUNICATE($k, \Delta^{k-1}, \boldsymbol{\theta}^k, \boldsymbol{f}^k$)

**end while**

**return** $\{\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k, \Delta^k\}$

**end function**

**Algorithm 6** Short term Decentralized Maintenance and Operations Algorithm

$\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \Delta\}_R \leftarrow$ DECENTDPOPT($\Delta_0, \boldsymbol{Q}^r_{relax}$)
$\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \Delta\} \leftarrow$ DECENTSGOPT($\Delta_R, \boldsymbol{Q}^r_{bin}$)

We use the regional solver to construct our decentralized algorithm represented in Algorithm 6 which comprises of two phases. In the first phase, we obtain convergence using a relaxation of the binary commitment and maintenance variables represented by the constraint set $Q^r_{relax}$. We utilize the phase and flow balance attained from the convergence of the relaxation to jump start the next phase involving binary commitment and flow variables represented by $Q^r_{bin}$. Such a two phased method has been demonstrated as one of the feasible methods for decentralized, mixed integer convergence in [39, 50].

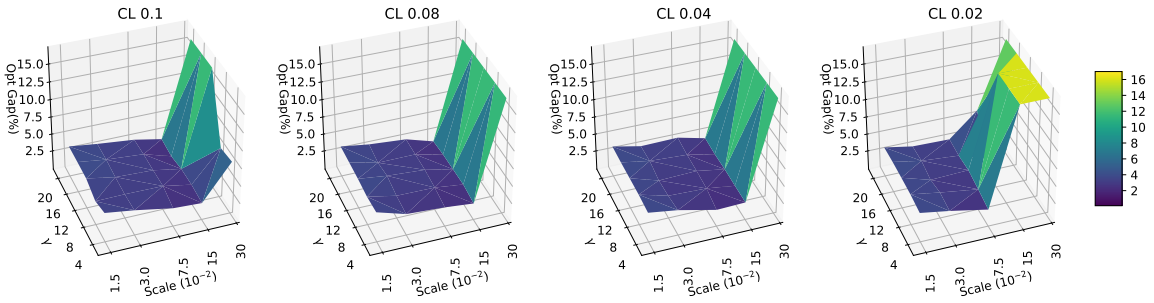



Figure 6.2: Robustness Analysis for 8 region decomposition

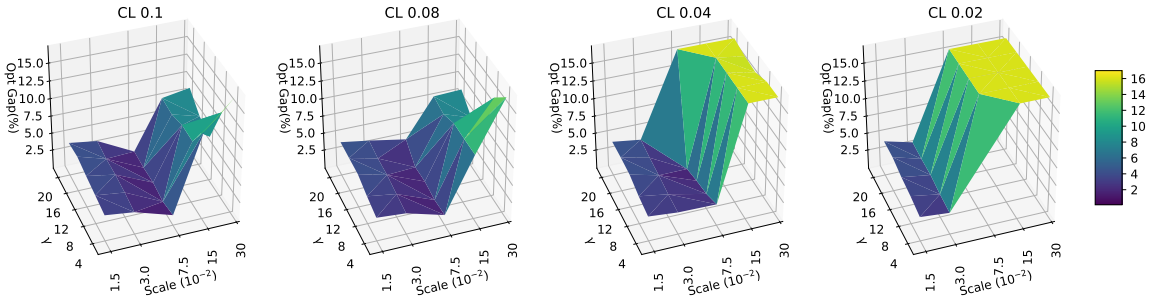



Figure 6.3: Robustness Analysis for 12 region decomposition

## 6.6 Results

In order to highlight the efficacy of our DP framework, we conduct numerous experiments on the 8 and 12 region decompositions of the IEEE 118 bus case. Our experiments revolve around analyzing the solution quality with respect to differing convergence limits, varying injected noise levels as well as changing the lookback window size for the control chart. We compute the solution quality relative to a centralized, non DP formulation of the maintenance problem.

Table 6.1: EWMA Mixing parameter $\eta$

| Scale | $\gamma$ | | | | |
|---|---|---|---|---|---|
| $(10^{-2})$ | 4 | 8 | 12 | 16 | 20 |
| 1.5 | 0.997 | 0.9976 | 0.9982 | 0.9988 | 0.9994 |
| 3.0 | 0.994 | 0.9952 | 0.9964 | 0.9976 | 0.9988 |
| 7.5 | 0.985 | 0.988 | 0.991 | 0.994 | 0.997 |
| 15.0 | 0.97 | 0.976 | 0.982 | 0.988 | 0.994 |
| 30.0 | 0.94 | 0.952 | 0.964 | 0.976 | 0.988 |

### 6.6.1 Experimental Setup

Our decentralized implementation is based on the Message Passing Interface (MPI) which is a popular paradigm for distributed memory computation in the field of High Performance Computing (HPC). In order to implement Algorithm 6 each region in our problem was assigned to a single MPI process. Using a distributed memory model like MPI for communication helps us evaluate the algorithm in an environment close to the *real-world*, where each region may represent individual participants of an ISO. Further, we impose an overlay network on top of the MPI layer which restricts a particular node to communicate only with the nodes representing the neighbors for its own region. The computational framework and software provided in this chapter can be used as validation tools for large scale real world implementations on a myriad of computational platforms.

We used `python` as the programming language for the framework. The `mpi4py` [19] package which is an MPI package for `python` was used to build the decentralized framework. We used `Gurobi 7.1` [20] for solving the MIQP problem represented by Problem (6.2) on each node. We evaluate our model on the IEEE 118 bus case with data derived from the `MATPOWER` library [22]. We simulate a geographically dispersed set of regions on a high performance cluster consisting of Intel Xeon CPUs with a clock rate of 2.80GHz with each core representing one region. Our planning horizon is of 1 week with hourly operational decisions and maintenance windows lasting 6 hours each. A preferred window for each generator scheduled for maintenance is provided as input to the local subproblem. We limit the maximum deviation from such preferences to at most 4 maintenance epochs. We impose a total runtime restriction of 10,600 secs for convergence. We compute the EWMA mixing parameter $\eta = 1.5\omega \times 10^{-3}$ as represented in Table 6.1.

The primal and dual tolerance limits for local convergence is given by $\beta_p, \beta_d = CL \cdot |\mathcal{B}_r| \cdot |T|$ respectively, where $CL$ is the convergence limit parameter. A higher $CL$ value could potentially yield faster convergence due to limited iterations that lead to reduced requirement for information exchange. Therefore, a higher $CL$ value might be a more

preferred option given a reasonable degree of solution quality.

### 6.6.2 Benchmark

In order to benchmark our result, we consider a centralized version of Algorithm 6 without Differential Privacy. We use the centralized benchmark to rate the performance of the decentralized algorithm by measuring the relative optimality gap

$$OptGap = \frac{|\xi^{decent} - \xi^{cent}|}{\xi^{cent}} \tag{6.16}$$

where $\xi^{decent}, \xi^{cent}$ represent the total objective value upon convergence for the decentralized and centralized case respectively. Optimality gap values of experiments which did not converge have been capped at 16%.

### 6.6.3 Robustness Analysis

Figures 6.2, 6.3 depict surface plots pertaining to performance of our DP framework subject to different convergence limits. Each surface plots tracks the solution quality in terms of optimality gap with varying values of $\gamma$ and noise levels.

From Figures 6.2, 6.3 we observe that our DP framework provides stable performance, yielding optimality gap values of less than 5% in most cases for both 8 and 12 region decompositions. We also observe that a higher $CL$ value does not necessarily come at the cost of poor optimality gap. Even with a high noise level, an appropriate selection of $\gamma$ might lead to an acceptable solution quality of around 5% and 6% in the 8 and 12 region case respectively with $CL$ value of 0.1. Further, as expected, the figures reveal that convergence becomes harder with higher noise levels and reduced $CL$ values. The figures also show that the number of regions in the network has a strong bearing on the convergence behavior and optimality gap. Such a behavior is expected since more regions imply a higher number of tie lines whose corresponding flow and phase angle values need

to be balanced leading to greater difficulty in convergence.

### 6.6.4 Noise Analysis

Figure 6.4 depicts the 2- norm difference between the real flow and DP based flow values upon convergence for 8 and 12 region cases. We notice that with increasing scale of the noise as given on the x-axis, the flow perturbation value keeps increasing as well indicating our framework's success in preserving flow privacy. Further, the noise magnitude for each scale value does not vary much between the 8 and 12 region cases signaling its independent nature as well as applicability to different real world instances.

### 6.6.5 Lookback Analysis

Figure 6.5 depicts the trends observed for lookback sizes of 10 and 20 for a convergence limit of 0.1 in terms of a bar plot. We also compare the optimality gap obtained with varying noise values to the no noise case without DP.

Overall, Figure 6.5 shows that our framework performs remarkably well with varying noise levels compared to the no noise case. Moreover, we also observe that a lower lookback size leads to greater variance in optimality gap values. However, on the other hand, a lower lookback directly implies lesser number of iterations, leading to fewer message exchanges.

An interesting behavior in Figure 6.5 is the decreasing optimality gap with increasing noise levels. This trend persists till $15 \times 10^{-2}$ and $7.5 \times 10^{-2}$ noise levels for the 8 and 12 region cases respectively before increasing for greater noise levels. Such a behavior can be attributed to the sensitivity of ADMM to the penalty parameter $\rho$. Methods aimed at tuning the penalty parameter have been proposed [73] for convex problems, however their applicability to Mixed Integer problems remains unexplored. Therefore, the penalty parameter was empirically chosen based on repeated trials with different values. Automatically adjusting the penalty parameter during run time for a finer performance of the Mixed Integer

Table 6.2: Computational Time (secs)

| Scale | 8 Regions | | 12 Regions | |
|---|---|---|---|---|
| ($10^{-2}$) | Mean | Std. Dev. | Mean | Std. Dev. |
| 1.5 | 1030.35 | 6.86 | 663.84 | 7.58 |
| 3.0 | 1023.20 | 14.18 | 665.88 | 7.12 |
| 7.5 | 1031.46 | 4.51 | 1990.82 | 2514.67 |
| 15.0 | 2240.02 | 3159.36 | 7052.10 | 4443.22 |
| 30.0 | 9217.68 | 3279.17 | 5516.14 | 4100.72 |

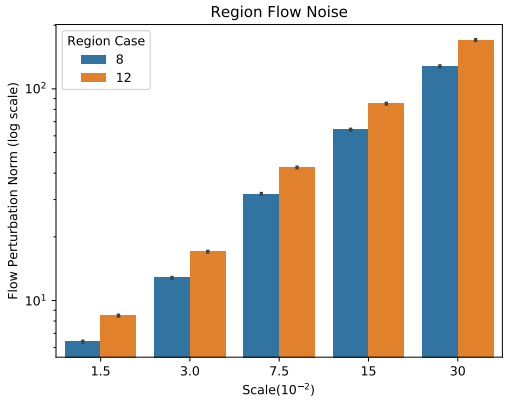



Figure 6.4: Flow Noise Analysis

formulation is a key component of our future work.

### 6.6.6 Computational Analysis

Table 6.2 shows the mean computational time along with the standard deviation for 8 and 12 region cases for all CL and $\gamma$ values. The figures in Table 6.2 also include cases where no convergence was observed within the maximum run time limit. We observe that for lower noise cases, the mean computation time proportionately decreases with increase in the number of regions from 8 to 12. Such a decrease is attributed to the computational speedup gained as a result of increased parallelism. However, we also observe an increase in the mean computational time for higher noise cases along with a variance. An increased variance and higher mean time to convergence is the consequence of two factors. First, due to a higher noise level, obtaining the Lagrangian balance might be more difficult leading to more iterations and prolonged convergence time. Additionally, higher noise levels result in

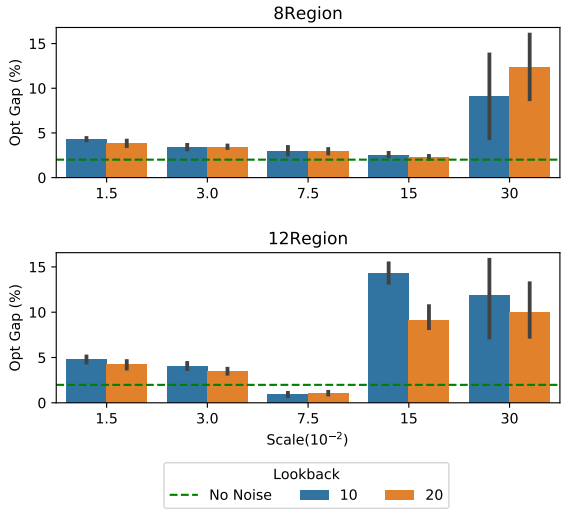



Figure 6.5: Computational Analysis

greater likelihood of not observing convergence within the prescribed time limit, ultimately increasing both the mean and variance.

## 6.7 Conclusion

In this chapter we present a differential privacy driven approach for solving planning problems in a decentralized fashion. We choose the short term maintenance and commitment problem as our target for demonstrating the efficacy of our approach due to its practical and critical aspects. Our decentralization is driven by a region based decomposition representing real world utility stakeholders. We obtain a decentralized formulation by dualizing the phase angles and flow constraints neighboring regions and iteratively balance these using ADMM. For orchestrating differential privacy, we exploit the linear relationship between the flow and phase angles. By injecting carefully coordinated exponential noise on the phase angles, we derive strong privacy guarantees on the flow values. In order to improve convergence, we adopt an EWMA based consensus averaging strategy in addition to a CLT based control chart. Our consensus strategy coupled with our control chart mechanism leads to a stable superior convergence for fairly large noise values. Further, using our HPC implementation of our decentralized framework, we show good solution quality that rivals

that of the centralized as well as the no noise benchmarks.

# CHAPTER 7
# BLOCKCHAIN BASED DECENTRALIZED CYBER ATTACK DETECTION FOR LARGE SCALE POWER SYSTEMS

## 7.1 Introduction

Affordable sensor and communication technologies have given rise to a growing wave of industrial digitization. The power industry has been at the forefront of this trend that has culminated into a digital transformation of the power grid. Such levels of digitization have given rise to automation and digital control components that are collectively referred to as Industrial Control Systems (ICS). Until recently, ICS used specialized communication and control protocols that made them relatively immune to cyberattacks. However, with increase in IIoT enabled assets, traditional ICS have gradually become heavily integrated with standard IT components. Meanwhile, on the physical level, the grid has evolved into a complex network with a high degree of interdependency among utility providers [74, 75]. Such an interdependent, digitized grid has an increased vulnerability to various kinds of cyberattacks [76]. Tackling these vulnerabilities requires competing utilities to share sensitive information with a trusted, centralized entity that can quickly assess cybersecurity related threats. The process of sharing data with a centralized entity can be challenging for many utilities due to the presence of a single point of failure, privacy concerns and competing market dynamics. Therefore, this chapter proposes a blockchain based, decentralized methodology for detecting the probability of a global network cyberattack while preserving data privacy. Our approach is particularly useful in the context of IIoT enabled assets that yield real time sensor data to help monitor the power network.

A significant portion of ICS-focused cyberattacks involve data manipulation. Often the intent of these attacks is to impact asset reliability either by accelerating physical or

efficiency degradation or causing sudden breakdowns. The *Stuxnet* worm has often been referenced as a classic example of an ICS-focused cyberattack with data manipulation [77]. Another popular example was the Aurora Generator Test in which a 2.25 MW substation generator was destroyed through a designed cyberattack that caused an out-of-sync closing of protective relays [78].

ICS cyberattacks involving data manipulation have been classified into three major types, false data injections [79], replay attacks [80] and covert attacks [81]. This chapter considers ICS replay attacks where a malicious agent replays sensor measurements representing normal operating conditions in order to mask underlying malicious control actions. We focus on investigating coordinated, large scale scenarios where an ICS replay attack is mounted on more than one regional utility provider in the power network. We refer to such attacks as global network cyberattack, or global attacks for short. ICS replay attacks are difficult to detect. Most of the existing detection algorithms do not distinguish between replay cyberattacks and naturally occurring equipment or controller faults. Furthermore, detection schemes are inherently based on some underlying hypothesis test, which means that there is always some level of unavoidable false alarms. Consequently, in large scale settings like the one considered in this chapter, detection schemes will tend to significantly overstate the cyber threat level. However, it is a well-known fact that false alarm rates increase with the number of hypotheses tests being conducted [82].

While current state-of-the-art ICS attack detection methods are intended for individual plant sites and assets, this chapter considers regional and global network attacks that require collaboration between the individual stakeholders. This collaborative setting involves "pooling" alarms from individual plants and assets, which tends to significantly reduce the false alarm rates. With current frameworks, such tasks, require competing stakeholders to share their data with a trusted third party which raises widespread privacy concerns. The U.S. Department of Energy operates the Cyber Risk Information Sharing Program (CRISP) [83] which provides utility members with a platform specifically designed to share sensitive

data. However, sharing data through a centralized repository presents several privacy risks along with efficiency and agility issues [84]. In the absence of legal obligation to participate, many critical utility providers do not pursue membership of data sharing cooperatives like CRISP. In addition, programs like CRISP also require hardware upgrades to existing IT infrastructure and associated costs as well [83].

In this chapter, we propose the use of the blockchain for detecting globally coordinated replay attacks in a decentralized fashion. Blockchains typically rely on a consensus among multiple mistrusting parties to achieve a consistent global state. As a result, blockchain based platforms are decentralized in nature and do not involve a centralized chain of command. Our framework therefore ensures data privacy by allowing individual utilities to run their own detection algorithms locally. This leads to full data ownership with complete data privacy and eliminates the associated cost of setting up a trusted, centralized third party. Moreover, we utilize blockchain driven Smart Contracts (SC) to estimate the likelihood of a global replay attack based on alarms and insights aggregated from the various utilities.

A decentralized computational strategy helps eliminate data privacy concerns as well as infrastructure costs of a centralized paradigm. However, in the absence of a versatile platform like the blockchain, an information diffusion mechanism would become essential for estimating the global cyber health status. Gossip protocols are an important type of diffusion technique aimed at estimating the global state of a system in a peer-to-peer fashion [85, 86]. Among the state-of-the-art, Broadcast Gossip (BG) is considered to be a scalable, computationally efficient gossip method that relies on neighbor based message exchanges in order to estimate the global state [86]. Therefore, from a computational standpoint, BG can also be employed to detect the likelihood of a global replay attack in the absence of a blockchain. As a result, in this chapter, we develop a BG based global replay attack detection detection framework for benchmarking purposes. We compare and contrast our blockchain driven approach against the BG in a theoretical as well as in an empirical

manner. Our results indicate that the blockchain provides a sound computational platform allowing for global aggregation of local outputs in a timely, accurate and reliable fashion. The major contributions of this work are summarized in detail below:

- We develop a decentralized mechanism that relies on Bayesian inference in order to detect a globally coordinated replay attack with full regional data privacy.
- We introduce Theorem 6, specifically geared towards maintaining computational efficiency of Bayesian inference on a blockchain platform.
- With the help of Theorem 6, we design a blockchain based framework for computing the global attack probability with only one global multiplication and addition steps.
- We propose the BG framework as a benchmark and theoretically compare its performance with the blockchain based approach with the help of Theorem 7.
- We implement our framework on an Ethereum based private blockchain network to demonstrate its scalability and applicability for varying degrees of cyber threat parameters.

Our results conclusively show that the blockchain driven framework is vastly superior to conventional, state-of-the-art information diffusion paradigms, like BG, both in terms of computational performance as well as accuracy of results.

## 7.2 Related Work

Cybersecurity of ICSs is a critical component of industrial application domain such as power systems [87, 76]. Attacks like DoS, DDoS, phishing, spoofing and eavesdropping that target generic IT systems can often be effectively detected and isolated by monitoring network traffic [88]. However, data manipulation attacks have little impact on ICS components [89]. Through manipulating controller/sensor data or even data at rest, attackers can damage critical assets through malicious control actions and incorrect state estimations

leading to degraded asset performance. Moreover, it has also been shown that data attacks can be designed to bypass basic verification methods relying on Cyclic Redundancy Check (CRC), User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). [90]

Numerous model-based detection frameworks have been proposed as an added layer of protection for ICS attacks involving data manipulation [79, 91]. Most detection algorithms rely on differences between actual measurements and those estimated by a model of the physical system. Differences between the estimated (or predicted) and observed measurements or states (i.e., residuals) can be used to detect possible cyberattacks. In most cases, a sequential goodness-of-fit testing procedure serves as the basis for detecting the attack. However, these approaches have been shown to be inefficient in detecting replay attacks, primarily because the observed measurements (replayed data) often match measurements estimated by the system's model. Several approaches to address this problem have been proposed in the literature with the most popular one relying on a type of private authentication. For example, the authentication methods proposed in [80] and [92] utilize calibrated control signals in the form of white noise to detect replay attacks. Specific noise signals are intentionally ingested into the system at designated time points. The correlation between the control signals and the residuals generated by the system model, a Kalman filter in this case, is used to detect potential attacks.

The use of blockchain has been proposed as a gateway to ensuring data privacy and security. The authors in [93] build a lightweight, private blockchain paradigm for enhancing the security and privacy of IIoT driven manufacturing platforms. Further, the work done in [94] utilizes a privacy preserving blockchain paradigm for orchestrating secure energy trading. Use of blockchain has been considered in power systems in numerous works in the recent past. Work done in [95] talks about the use of blockchain to establish a decentralized, secure technique for transactive energy. The authors in [96] propose using a permissioned blockchain to achieve security, fairness and overall balance with respect to energy trading in vehicular energy networks. The work done in [97] proposes the use of the blockchain to

handle and trace back energy losses in microgrids that incorporate PV nodes. Authors in [98] propose a blockchain based data protection scheme for smart meters as a mechanism to protect against false data injection.

Therefore, blockchain driven approaches are being considered as the perfect computational platforms for collaboratively detecting attacks and anomalies on a global scale with full data privacy [99]. On the other hand, despite its immense potential, efforts exploring the use of the blockchain towards detecting globally coordinated replay attacks on power network ICSs largely remains understated.

### 7.3 Problem Formulation

We primarily use the blockchain as a computational platform for aggregating outputs from local replay attack detection algorithms. In developing our blockchain based global attack detection framework, we consider a power network that is divided topologically into a set of distinct regions denoted by $\mathcal{R} = \{1, 2, 3 \dots n\}$. Each region can be thought of as a utility provider with multiple power plants.

Our problem formulation can be viewed as having two components, a local and a global component. The local component represents regional plants that belong to a single utility provider. We assume that each region executes the local algorithms aimed at detecting replay attacks on their plants' ICS. We refer to the local component as the *regional detection model*. The global component, i.e. the *network detection model*, concerns the detection of a coordinated global attack at the network level. A blockchain architecture is used to assess the global state of the entire network based on regional insights that are aggregated in a decentralized, privacy preserving manner. The aggregation determines the overall probability of a global network attack. This model assumes that all regions are equally likely to experience a replay attack which are mutually independent. Independence is assumed here for mathematical convenience. More importantly, a global network attack is assumed to occur when at least two regions report local attacks.

### 7.3.1 Regional Detection Model

For the regional detection model, we consider a single region $i$ comprised of $p$ generators monitored and controlled by a single ICS where the state of each generator can be represented by $m$ variables. We assume that the dynamics of this system of generators can be characterized by a linear time-invariant (LTI) model described by Equations (7.1), (7.2).

$$x_{t+1} = Ax_t + Bu_t + v_t, \tag{7.1}$$

$$y_t = Cx_t + w_t \tag{7.2}$$

In Equations (7.1), (7.2), $x_t \in \mathbb{R}^{mp}$ represents the unobserved state of the system at time $t$, $u_t$ is the control action at time $t$, $y_t \in \mathbb{R}^{mp}$ denotes the sensor measurements at time $t$, which are assumed to be noisy realizations of the state. $v_t \in \mathbb{R}^{mp}$ and $w_t \in \mathbb{R}^{mp}$ represent process and measurement noise at time $t$, respectively. Although this modeling framework has limitations, it has still been used extensively in the literature [80].

In this setting, the Kalman filter is known to be the optimal state estimator[100]. The residuals $r_t$ are defined as the differences between the actual measurements $y_t$ and the predicted measurements $C\hat{x}_{t|t-1}$ as estimated by the Kalman filter. The control action $u_t$ is calculated using a linear-quadratic Gaussian (LQG) controller, which is the optimal controller under the LTI setting [101]. The Kalman filter and the LQG controller are estimated based on Equations (7.3)-(7.6).

$$\hat{x}_{t|t-1} = A\hat{x}_{t-1|t-1} + Bu_{t-1} \tag{7.3}$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + Kr_t \tag{7.4}$$

$$r_t = y_t - C\hat{x}_{t|t-1} \tag{7.5}$$

$$u_t = L\hat{x}_{t|t} \tag{7.6}$$

In Equation (7.6), $L = -(B^TSB + U)^{-1}B^tSA$ and $S = A^TSA + W - A^TSB(B^TSB +$

$U)^{-1}B^TSA$ is the solution to the Riccati equation [101].

Now consider a system under a replay attack. Let $y_t^r$ denote sensor measurements corresponding to normal operations, and $u_t'$ and $y_t'$ denote the manipulated control actions and sensor measurements, respectively. A replay attack at time $t$ requires that $u_t' = u_t + a$ and $y_t' = y_t^r$. As shown in [102], replay attacks can be detected by monitoring the covariance matrix of the residuals $r_t$. Using this approach, let $\sigma^i$ be a random variable representing the probability of an alarm triggered by region $i$. That is, $\sigma^i = 1$ if a replay attack is detected and $\sigma^i = 0$ otherwise. We define $\hat{\sigma}^i$ as the ground truth that represents whether a replay attack is indeed underway. There are two classic errors that can occur in this setting. Type-I error, $\alpha$, represents the probability of a false alarm, i.e., the algorithm triggers an alarm when there is no attack. The Type-II error, $\beta$, represents the probability of a false negative, i.e., where the detection algorithm fails to detect a true replay attack. These errors can be defined more formally in terms of a region $i$ as follows.

$$\alpha_i = Pr(\sigma^i = 1 | \hat{\sigma}^i = 0) \tag{7.7}$$

$$\beta_i = Pr(\sigma^i = 0 | \hat{\sigma}^i = 1) \tag{7.8}$$

In Equations (7.7), (7.8) $\hat{\sigma}^i = 1$ in the event the system is truly under a replay attack whereas $\hat{\sigma}^i = 0$ otherwise.

### 7.3.2 Network Detection Model

Consider a power network comprised of $n$ regions each reporting an alarm based on their local belief of an attack. Recall that a global network attack is triggered if there are two or more distinct regional alarms, i.e. at least two regions detect an attack. Let the set $S = \{\boldsymbol{s}_0, \boldsymbol{s}_1 \ldots \boldsymbol{s}_n\}$ denote all the scenarios that represent no global network attack, where $\boldsymbol{s}_0 \in \{0\}^n$ indicates a scenario where no regional alarms have been triggered and $\boldsymbol{s}_i \in \{0,1\}^n$ denotes scenarios where only one region $i$ triggers an alarm. Consequently, for a

scenario $\boldsymbol{s}_k$, $k > 0$, only its $k^{th}$ element $s_k^k = 1$. Next, let $\boldsymbol{\sigma} \in \{0,1\}^n = [\sigma^1, \sigma^2 \dots \sigma^n]$ be a vector of random variables representing regional alarm events. Given $\boldsymbol{\sigma}$, the probability of no global network attack can be expressed as $Pr(S|\boldsymbol{\sigma})$. Similarly, $Pr(\boldsymbol{\sigma}|\boldsymbol{s}_k)$ defines the probability of observing $\boldsymbol{\sigma}$ given a scenario $\boldsymbol{s}_k$ Since regional attacks are assumed to be independent, the following expressions hold true: (7.9)-(7.12).

$$Pr(\boldsymbol{s}_k) = \prod_{i=1}^{n} Pr(s_k^i) \tag{7.9}$$

$$Pr(\boldsymbol{\sigma}|\boldsymbol{s}_k) = \prod_{i=1}^{n} Pr(\sigma^i|s_k^i) \tag{7.10}$$

$$Pr(S|\boldsymbol{\sigma}) = \sum_{\boldsymbol{s}_k \in S} \frac{Pr(\boldsymbol{\sigma}|\boldsymbol{s}_k)Pr(\boldsymbol{s}_k)}{Pr(\boldsymbol{\sigma})} \tag{7.11}$$

$$Pr(\boldsymbol{\sigma}) = Pr(\sigma^1 \cap \sigma^2 \dots \cap \sigma^n) = \prod_{i=1}^{n} Pr(\sigma^i) \tag{7.12}$$

The occurrence of the event denoted by $s_k^i$ is only relevant in the context of the ground truth $\hat{\sigma}^i$. Therefore, we can state that $Pr(s_k^i) = Pr(\hat{\sigma}^i = s_k^i)$, where $Pr(\hat{\sigma}^i = s_k^i)$ reflects the prior probability of the existing ground truth $\hat{\sigma}^i$ being equal to $s_k^i$. Consequentially, substituting Equations (7.9) and (7.10) in (7.11), we obtain the relation represented by (7.13).

$$\sum_{\boldsymbol{s}_k \in S} Pr(\boldsymbol{\sigma}|\boldsymbol{s}_k)Pr(\boldsymbol{s}_k) = \sum_{\boldsymbol{s}_k \in S} \prod_{i=1}^{n} Pr(\sigma_i|\hat{\sigma}^i = s_k^i)Pr(\hat{\sigma}^i = s_k^i) \tag{7.13}$$

We also know that;

$$Pr(\sigma^i) = Pr(\sigma^i|\hat{\sigma}^i = 0)Pr(\hat{\sigma}^i = 0) + Pr(\sigma^i|\hat{\sigma}^i = 1)Pr(\hat{\sigma}^i = 1) \tag{7.14}$$

We note that $Pr(\sigma^i)$ and $Pr(\sigma_i|s_k^i)Pr(\hat{\sigma}^i = s_k^i)$ in Equations (7.13), (7.14), respectively, can be computed locally allowing us to propose the the following theorem.

**Theorem 6.** *The probability of no global attack denoted by $Pr(S|\sigma)$ is governed by the*

*equality,*

$$Pr(S|\boldsymbol{\sigma}) = \Big(\prod_{i=1}^{n} \frac{a_i}{Pr(\sigma^i)}\Big)\Big(1 + \sum_{i=1}^{n} \frac{b_i}{a_i}\Big)$$

*where,*

$$a_i = Pr(\sigma^i|\hat{\sigma}^i = 0)Pr(\hat{\sigma}^i = 0),\ b_i = Pr(\sigma^i|\hat{\sigma}^i = 1)Pr(\hat{\sigma}^i = 1)$$

*Proof.* Based on our assumptions, we derive the relations represented by Equations (7.15), (7.16).

$$Pr(\boldsymbol{\sigma}|\boldsymbol{s}_0)Pr(\boldsymbol{s}_0) = \prod_{i=1}^{n} Pr(\sigma^i|\hat{\sigma}^i = 0)Pr(\hat{\sigma}^i = 0) \tag{7.15}$$

$$Pr(\boldsymbol{\sigma}|\boldsymbol{s}_k)Pr(\boldsymbol{s}_k) = \prod_{i=1}^{n} Pr(\sigma^i|\hat{\sigma}^i = s_k^i)Pr(\hat{\sigma}^i = s_k^i) \tag{7.16}$$

Dividing Equations (7.16) with (7.15), we get

$$\psi_k = \frac{Pr(\boldsymbol{\sigma}|\boldsymbol{s}_k)Pr(\boldsymbol{s}_k)}{Pr(\boldsymbol{\sigma}|\boldsymbol{s}_0)Pr(\boldsymbol{s}_0)} = \prod_{i=1}^{n} \frac{Pr(\sigma^i|\hat{\sigma}^i = s_k^i)Pr(\hat{\sigma}^i = s_k^i)}{Pr(\sigma^i|\hat{\sigma}^i = 0)Pr(\hat{\sigma}^i = 0)} \tag{7.17}$$

Since, $s_k^i = 1$ only when $i = k$, we can rewrite Equation (7.17) as

$$\psi_k = \Big[\prod_{\substack{i=1 \\ i\neq k}}^{n} \frac{Pr(\sigma^i|\hat{\sigma}^i = 0)Pr(\hat{\sigma}^i = 0)}{Pr(\sigma^i|\hat{\sigma}^i = 0)Pr(\hat{\sigma}^i = 0)}\Big] \frac{Pr(\sigma^k|\hat{\sigma}^k = 1)Pr(\hat{\sigma}^k = 1)}{Pr(\sigma^k|\hat{\sigma}^k = 0)Pr(\hat{\sigma}^k = 0)} \tag{7.18}$$

Equations (7.17), (7.18) imply that,

$$Pr(\boldsymbol{\sigma}|\boldsymbol{s}_k)Pr(\boldsymbol{s}_k) = Pr(\boldsymbol{\sigma}|\boldsymbol{s}_0)Pr(\boldsymbol{s}_0)\frac{b_k}{a_k}, \quad \forall k > 0 \tag{7.19}$$

Recall from Equation (7.13), that

$$Pr(\boldsymbol{\sigma}|\boldsymbol{s}_0)Pr(\boldsymbol{s}_0) = \prod_{i=1}^{n} Pr(\sigma^i|\hat{\sigma}^i = 0)Pr(\hat{\sigma}^i = 0) = \prod_{i=1}^{n} a_i \tag{7.20}$$

Since $a_k, b_k$ can be purely computed by region $k$ we obtain Equation (7.21) by combining

Equations (7.19) and (7.20)

$$\sum_{\boldsymbol{s}_k \in S} Pr(\boldsymbol{\sigma}|\boldsymbol{s}_k) Pr(\boldsymbol{s}_k) = \Big(\prod_{i=1}^{n} a_i\Big)\Big(1 + \sum_{i=1}^{n} \frac{b_i}{a_i}\Big) \tag{7.21}$$

Based on Equations (7.21) and (7.12), we obtain

$$Pr(S|\boldsymbol{\sigma}) = \frac{\sum\limits_{\boldsymbol{s}_k \in S} Pr(\boldsymbol{\sigma}|\boldsymbol{s}_k) Pr(\boldsymbol{s}_k)}{\prod\limits_{i=1}^{n} Pr(\sigma^i)} = \Big(\prod_{i=1}^{n} \frac{a_i}{Pr(\sigma^i)}\Big)\Big(1 + \sum_{i=1}^{n} \frac{b_i}{a_i}\Big)$$

□

The entities $a_i, b_i$ can be computed in a simplified manner as follows.

$$a_i = Pr(\sigma^i, \hat{\sigma}^i = 0) = Pr(\sigma^i|\hat{\sigma}^i = 0) Pr(\hat{\sigma}^i = 0) \tag{7.22}$$

$$b_i = Pr(\sigma^i, \hat{\sigma}^i = 1) = Pr(\sigma^i|\hat{\sigma}^i = 1) Pr(\hat{\sigma}^i = 1) \tag{7.23}$$

Moreover, the prior distribution can be updated in a purely local fashion using:

$$Pr(\hat{\sigma}^i = 0) = \frac{a_i}{a_i + b_i} \tag{7.24}$$

$$Pr(\hat{\sigma}^i = 1) = \frac{b_i}{a_i + b_i} \tag{7.25}$$

Theorem 6 indicates that the global attack probability can be computed through one global multiplication and addition of the two terms, $\frac{a_i}{Pr(\sigma^i)}, \frac{b_i}{a_i}$. We therefore incur significant computational benefits especially on a blockchain based framework where computation is expensive.

## 7.4 Blockchain Based Framework for Global Replay Attack Detection

The introduction of Smart Contracts (SCs) have been an important addition to the blockchain paradigm. An SC is typically a snippet of code that resides on the blockchain. It can contain

complex program logic on the blockchain and can be invoked by any party having access to the blockchain. Once invoked, an SC is self-triggering and proceeds to alter the state of the ledger with the help of the underlying consensus protocol. As a result, an SC can be used for executing business logic through consensus among mistrusting parties paving the way for a decentralized application. Therefore, due to their versatility, blockchain driven Smart Contracts (SC) provide an ideal environment for the Network Detection model in a fully decentralized fashion.

Solidity [1] is a popular SC oriented programming language that can be leveraged for developing highly versatile decentralized applications. A key constraint of languages like Solidity is the lack of floating point arithmetic operations [2]. This is done to primarily reduce the computational burden on the underlying consensus protocols. As a result, one has to define a precision conversion factor for converting floating point values to integers before feeding them as inputs to the SC on the front end side. Obviously, the choice of the precision factor could have a wide impact on the detection accuracy of a global replay attack. In this section, we propose a Solidity based Smart Contract(SC) design that embodies our global replay attack detection paradigm discussed in Section 7.3.

### 7.4.1 Smart Contract Design

Table 7.1, 7.2 depict the attributes and functions that form an integral part of the SC design for our framework. In Table 7.1, the precision factor is denoted by $D$ and arrays $aOPs, bOa$ store the integer values $\frac{Da_i}{Pr(\sigma^i)}, \frac{Db_i}{a_i}$ of all regions respectively. In Table 7.2, the functions $updateData$, $aggregateValues$ can be asynchronously invoked by any region for updating the SC with their local values and obtaining the corresponding global aggregates respectively.

[1] https://github.com/ethereum/solidity

[2] https://solidity.readthedocs.io/en/v0.5.3/types.html#fixed-point-numbers

Table 7.1: Solidity based Smart Contract Attributes

| Attribute | Type | Description |
|---|---|---|
| *D* | *uint256* | Precision for floating point conversion. |
| *n* | *uint* | The total number of regions i.e. $\|\mathcal{R}\|$ |
| *aOPs*[] | *uint256* | Array for storing $\frac{Da_i}{Pr(\sigma^i)}, \forall i \in \mathcal{R}$ |
| *bOa*[] | *uint256* | Array for storing $\frac{Db_i}{a_i}, \forall i \in \mathcal{R}$ |

Table 7.2: Solidity based Smart Contract Function

| Function | Invoker | Description |
|---|---|---|
| *updateData* | *Region i* | sets $aOPs[i]$, $bOa[i]$ |
| *aggregateValues* | *Region i* | returns $\prod_{j=1}^{n} aOPs[i]$, $\sum_{j=1}^{n} bOa[i]$ |

### 7.4.2 Blockchain Based Global Attack Detection Algorithm

Algorithm 7 presents the details of blockchain based global replay attack detection which is executed in a decentralized fashion. We assume an off chain interaction informs all the regions about the SC address, total number of regions as well as the precision factor. Every region initially determines its corresponding local alarm value $\sigma^i$. Let $x_i = \frac{a_i}{Pr(\sigma^i)}, y_i = \frac{b_i}{a_i}$ represent the local statistical values which are converted to integers using the precision factor $D$ and pushed to the SC by invoking $SC.updateData$. Any region can asynchronously invoke $SC.aggregateValues$ in order to obtain $x^b = \prod_{i=1}^{n} Dx_i$ and $y^b = \sum_{i=1}^{n} Dy_i$. A global estimate of no attack can then be computed locally using Theorem 6. A complement of the result can be used to infer the presence of a global attack. At each epoch, the prior distribution pertaining to local alarm values gets updated locally.

## 7.5 Performance Comparison of Blockchain and BG driven approaches

For developing a decentralized global replay attack detection, we assume the existence of an underlying connectivity graph that represents the system level interconnection among utilities [74]. In this graph, each vertex represents a utility or region. An edge exists if there is a shared transmission line between the corresponding utilities. Without a central

**Algorithm 7** Decentralized blockchain based algorithm

owner region deploys and initializes $SC$ on blockchain
**for** i=1,2,3...n **in parallel do**
  initialize $D, n$ and obtain $SC$ address.
  **while** true **do**
    determine the message $\sigma^i$
    compute $Pr(\sigma^i)$ using Equation (7.14)
    compute $x_i, y_i$ using Equations (7.22),(7.23)
    invoke $SC.updateData\Big(Dx_i, Dy_i, i\Big)$
    $x^b, y^b \leftarrow SC.aggregateValues()$
    using Theorem 6 compute $1 - Pr(S|\boldsymbol{\sigma})$ such that,
    $$Pr(S|\boldsymbol{\sigma}) = x^b(D + y^b)/D^{(n+1)}$$
    update prior distribution using Equations (7.24),(7.25)
  **end while**
**end for**

aggregator, a diffusion mechanism must take place over the existing connectivity graph in order to detect globally coordinated replay attacks. Therefore, we develop a novel diffusion algorithm based on state-of-the-art BG as a benchmark strategy for our blockchain based approach [86]. BG can be used to compute the global average of values held by the vertices of the connectivity graph. BG avoids the computational bottlenecks found in other gossip protocols while converging to the global average in expectation [86]. Being inherently decentralized, consensus driven and peer-to-peer in nature, BG forms the ideal benchmark for comparing the performance of a blockchain driven framework.

### 7.5.1 BG based Reformulation of Global Attack Detection

Since the total number of agents in the system is known, BG can be used to estimate the global sum and product terms present in Theorem 6 as well. Recall $x_i, y_i$, $\forall i \in \mathcal{R}$ from Algorithm 7. For computing the sum $y$, the reformulation is trivial and consists of estimating the global sum from the global average $\sum_{k=1}^{n} y_i/n$. In order to use BG to calculate the product $x = \prod_{i=1}^{n} x_i$, we reformulate $x = e^u$, where $u = \sum_{i=1}^{n} u_i$, and $u_i = \log x_i$. We leverage the BG protocol to compute $\sum_{i=1}^{n} u_i/n$ which can be used to estimate $x$ in a purely

peer-to-peer fashion.

### 7.5.2 Performance Analysis

We wish to compare and contrast the precision factor based error introduced in the blockchain framework against the expected asymptotic error in a BG driven framework. As a result, we propose Theorem 7 which helps characterize the conditions favorable for the BG to remain competitive with our blockchain framework. For Theorem 7, we consider a set of agents connected by graph $G$ with Laplacian $L$. For our analysis of Theorem 7, we assume a constant value of $\beta$ across all regions. Further, we let $\lambda_{n-2}(L)$ and $\lambda_1(L)$ denote the second smallest and the largest eigenvalues of $L$ where $\gamma \in (0, 1)$ represents the mixing parameter [86]. The mixing parameter dictates the contribution of values recieved from each neighbor during the BG protocol at each vertex [86]. As illustrated in [86], the expected BG error bound for computing the average $\bar{z} = \sum_{i=1}^{n} z_i/n$ where $z_i$ is the local value held by agent $i \in \{1, 2, \dots n\}$ is given by

$$\lim_{t\to\infty} \Delta(z^t)^2 \leq (\Delta z^0)^2(1-r), \text{ where } r = \frac{\gamma \frac{\lambda_{n-2}}{\lambda_1}}{1 - \frac{1-\gamma}{2n}\lambda_1} \tag{7.26}$$

**Theorem 7.** *Let $\Delta p_b, \Delta p_g^{\infty}$ denote the precision error induced in a blockchain framework and the limiting asymptotic mean square error from BG respectively. $\Delta p_g^{\infty} \leq \Delta p_b$ if*

$$D \leq \frac{1}{\beta\sqrt{n(1-r)\Big[\Big(\frac{\Delta x^0}{x}\Big)^2 + \Big(\frac{\Delta y^0}{1+y}\Big)^2\Big]}}$$

*where $x^0, y^0$ represent the initial error of $x, y$ respectively in case of BG.*

*Proof.* We consider the error on $p = x \cdot (1 + y)$ to obtain

$$\Delta p^2 = p^2\Big[\Big(\frac{\Delta x}{x}\Big)^2 + \Big(\frac{\Delta y}{1+y}\Big)^2\Big] \tag{7.27}$$

Analyzing error on $x, y$, we have

$$\Delta x^2 = |x|^2\Big[\sum_{i=1}^{n}\Big(\frac{\Delta x_i}{x_i}\Big)^2\Big],\ \Delta y^2 = \sum_{i=1}^{n}(\Delta y_i)^2$$

With a precision factor of $D$, we know that $\Delta x_i, \Delta y_i \leq \frac{1}{D}$ leading to the relations

$$\Big(\frac{\Delta x}{x}\Big)^2 \leq \frac{1}{D^2}\sum_{i=1}^{n}\Big(\frac{1}{x_i}\Big)^2,\ \Big(\frac{\Delta y}{1+y}\Big)^2 \leq \Big[\frac{n}{(1+y)D}\Big]^2$$

We know that,

$$a_i = Pr(\sigma^i|s^i = 0)Pr(s^i = 0) \implies x_i = Pr(s^i = 0|\sigma^i)$$

$$x_i = \begin{cases} \beta, & \text{when } \sigma_i = 1 \\ 1-\alpha, & \text{when } \sigma_i = 0 \end{cases}$$

Let $p_b$ denote the global probability of no attack obtained with a blockchain based framework with precision factor $D$. Since $\beta, \alpha \approx 0$ and $\frac{1}{x_i} \leq \frac{1}{\beta}$, we obtain

$$\Big(\frac{\Delta p_b}{p_b}\Big)^2 \leq \frac{1}{D^2}\Big[\frac{n}{\beta^2} + \Big(\frac{n}{1+y}\Big)^2\Big]$$

Since $\frac{1}{(1+y)^2} \leq 1$, we can safely assume that $\frac{1}{(1+y)^2} \in O(\frac{1}{n\beta^2})$ thereby leading to the following error estimate for the blockchain based algorithm

$$\Big(\frac{\Delta p_b}{p_b}\Big)^2 \leq \Big(\frac{n}{D}\Big)^2 \cdot \frac{1}{n\beta^2} \tag{7.28}$$

On the other hand for the BG algorithm recall that $x = e^u$, where $u = \sum_{i=1}^{n} u_i$, $u_i = \log x_i$. Based on error analysis we obtain

$$(\Delta x)^2 = \Big(\frac{\partial x}{\partial u}\Delta u\Big)^2 \implies \Big(\frac{\Delta x}{x}\Big)^2 = \Delta u^2$$

Let $p_g$ denote the global probability of no attack obtained from a BG based framework. We can therefore say that,

$$\Big(\frac{\Delta p_g}{p_g}\Big)^2 \leq \Delta u^2 + \Big(\frac{\Delta y}{1+y}\Big)^2 \tag{7.29}$$

Substituting the expected upper bound on error terms $\Delta u^2, \Delta y^2$ from Equation (7.26), we obtain

$$\Big(\frac{\Delta p_g}{p_g}\Big)^2 \leq n^2(1-r)\Big[(\Delta u^0)^2 + \Big(\frac{\Delta y^0}{1+y}\Big)^2\Big] \tag{7.30}$$

Based on Equations (7.28) and (7.30), for the BG algorithm to completely outperform the blockchain based method, the following condition must be satisfied.

$$n^2(1-r)\Big[(\Delta u^0)^2 + \Big(\frac{\Delta y^0}{1+y}\Big)^2\Big] \leq \Big(\frac{\Delta p_b}{p_b}\Big)^2 \leq \Big(\frac{n}{D}\Big)^2 \cdot \frac{1}{n\beta^2}$$

Since we know that $\Delta u^0 = \frac{x^0}{x}$ we can state that

$$D \leq \frac{1}{\beta\sqrt{n(1-r)\Big[\Big(\frac{\Delta x^0}{x}\Big)^2 + \Big(\frac{\Delta y^0}{1+y}\Big)^2\Big]}} \tag{7.31}$$

□

Theorem 7 shows us that the precision factor $D$ has an inverse relation to the initial BG error, $\Delta x^0/x + \Delta y_0^2/y$. It can also be noted that $r(\gamma)$ is a monotonously increasing function with $0 \leq r(\gamma) \leq \frac{\lambda_{n-2}}{\lambda_1}$ [86]. Therefore, Theorem 7 indicates a direct relationship between the precision factor $D$ and the mixing parameter $\gamma$ as well as the eigenvalue ratio denoted by $\frac{\lambda_{n-2}}{\lambda_1}$.

Based on observations from Theorem 7, we can postulate several constraints on BG in order for it to match the detection quality of a blockchain driven approach with a high precision factor $D$. First, BG must preferably start with low initial error with respect to the global values of $x, y$. Second, since $\gamma \to 1$ is more favorable for BG, only scant perturbation of the local estimate can be allowed when new neighbor messages are received.

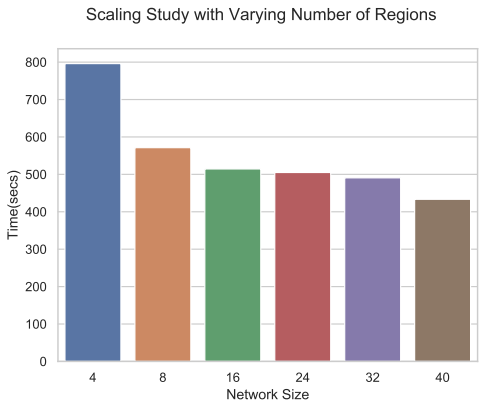



Figure 7.1: Computational Scaling Study comparing Simulation Time against varying number of total regions for IEEE 3012 bus case

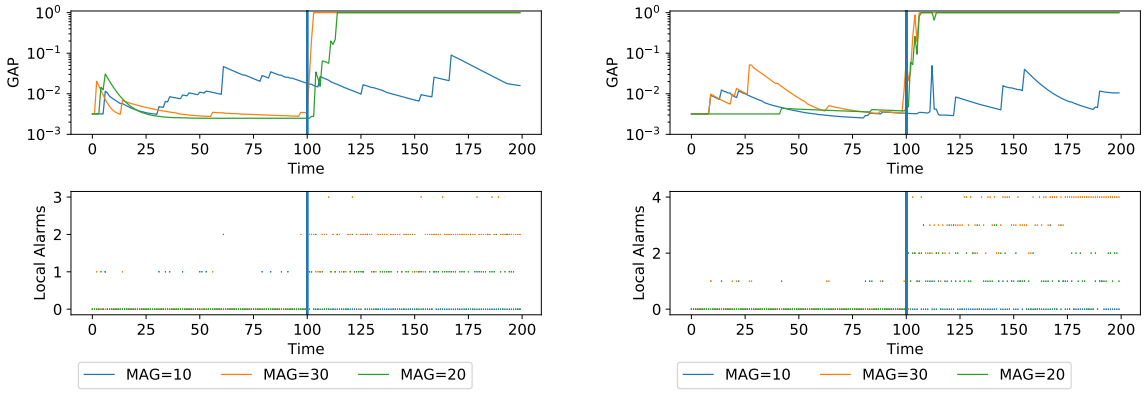
Therefore, a BG framework cannot afford a drastic change in the overall network mean, making a low initial error state on all nodes imperative to its success. Lastly, for good detection quality, the BG demands a highly connected underlying graph as indicated by the requirement for $\frac{\lambda_{n-2}}{\lambda_1} \rightarrow 1$.

The prerequisite constraints for the BG to outperform the blockchain present significant implementation challenges especially in a large scale power network with rapidly evolving global cyber health status. As a result, a blockchain driven framework is a highly favorable option for delivering an accurate, reliable and timely estimate of the global attack probability.

## 7.6 Experimental Results

Our blockchain results utilize a Geth[3] based, Ethereum private blockchain network orchestrated on a 2.7 GHz Intel Xeon processor with 128 cores. We used *Solidity 5.2* to compile the SC and utilized multithreading to simulate the region processes. Each region is assigned one thread each for the Ethereum node and the regional detection model respectively. We used the IEEE 3012 bus transmission network to simulate a large scale transmission network consisting of 150 generators. In our experiments, we set $\alpha = 0.005$ for the regional

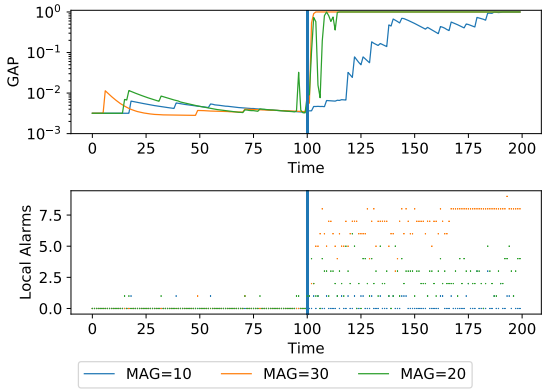
[3]https://geth.ethereum.org



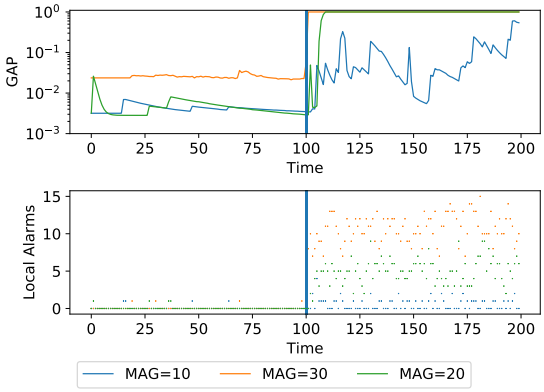

(a) At most 2 regions under attack




(b) At most 4 regions under attack




(c) At most 8 regions under attack

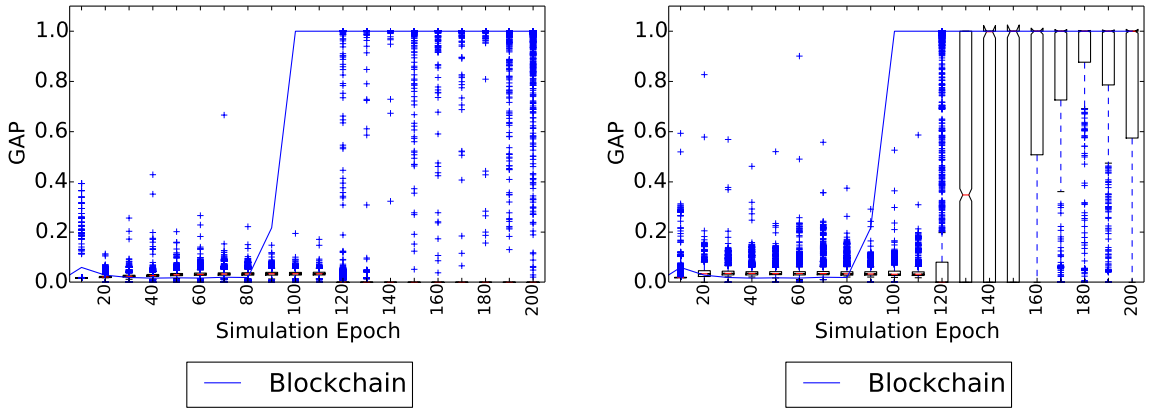



(d) All 16 regions under attack

Figure 7.2: 16 region decomposition of IEEE 3012 bus case : Global Attack Probability detection with varying number of regions under attack

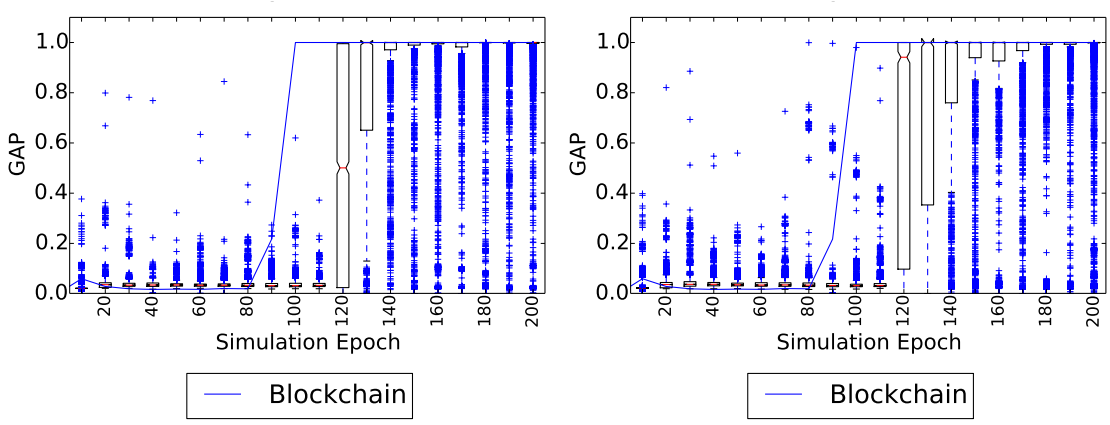
detection component for all regions. For our blockchain framework, we use a precision factor value of $D = 10^6$. Further, we employ the Proof-of-Authority (PoA) consensus protocol [4] given its suitability for power system applications wherein utilities can act as the authorities. PoA eliminates computationally intensive consensus by shifting the onus of trust to the authorities themselves. Our simulations occur at discrete epochs at which local alarms are reported. In all our experiments the simulation horizon is 200 epochs.

[4]https://wiki.parity.io/Proof-of-Authority-Chains




(a) 1 gossip round per epoch

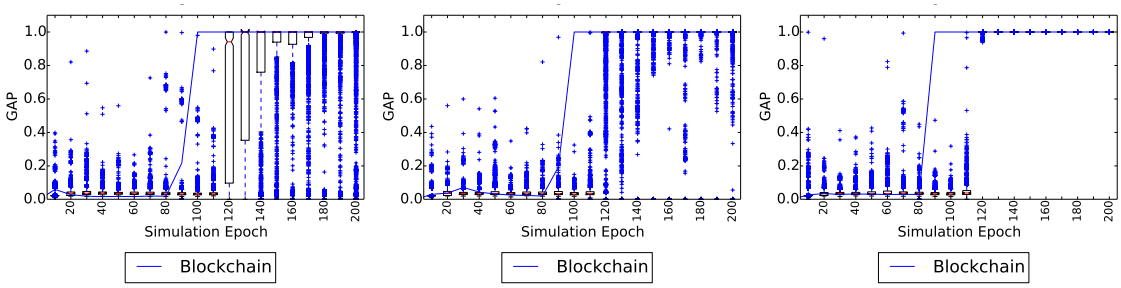
(b) 10 gossip rounds per epoch




(c) 25 gossip rounds per epoch

(d) 50 gossip rounds per epoch

Figure 7.3: 40 region decomposition of IEEE 3012 bus case : Comparison with BG with varying local gossip rounds with fixed 2 regions under attack




(a) At most 2 regions under attack (b) At most 4 regions under attack (c) At most 8 regions under attack

Figure 7.4: 40 region decomposition of IEEE 3012 bus case : Comparison with BG with varying number of regions under attack with local gossip rounds fixed at 50

### 7.6.1 Computational Performance and Accuracy

We evaluate the computational performance of our blockchain framework by considering varied regional decompositions of the IEEE 3012 bus case. Figure 7.1 presents the incurred time for simulation with varying number of total regions for our blockchain based framework. We observe that a lesser number of regions takes longer to simulate in general. We also notice that the overall simulation time decreases sharply after 4 regions, decreasing gradually and eventually settling at approximately 500secs. This is because as a result of the Clique [5] PoA consensus protocol, lower values of $n$ contribute to a high overall simulation time due to increased block generation time [103]. However, increasing $n$, results in lesser block generation time but more communication burden. Therefore, we observe that the simulation time eventually stabilizing around a fixed value even as $n$ is increased.

Figure 7.2 presents results pertaining to experiments involving varied number of regions under attack along with varying degree of attack magnitude. We observe that the framework is robust to varying attack magnitudes and is able to qualitatively detect attacks regardless of attack magnitude. Further, the results also indicate that the framework performs consistently with varying number of regions under attack as well.

### 7.6.2 Evaluation against BG

Figures 7.3 and 7.4 depict box and line plots against discrete simulation epochs corresponding to the BG and the blockchain respectively for a magnitude of 20. Each box plot presents the spectrum of global probability values recorded by all region across 100 repeated trials of BG. Similarly, the line plot represents the global probability values determined by the blockchain framework.

Figure 7.3 compares the performance of the blockchain version with a BG based scheme with varying number of gossip rounds for each simulation epoch by enforcing two regional attacks. We observe that the blockchain based framework detects a global attack faster

[5] https://github.com/ethereum/EIPs/issues/225

than the BG by a margin of 30 epochs. Moreover an increasing number of BG rounds per epoch leads to a sharp decrease in the mean, median and the variance of the global attack probability across all regions. Figure 7.3 demonstrates that despite 50 synchronous gossip rounds per epoch, a global attack detection cannot be successfully disseminated among all regions in a BG framework.

Figure 7.4 compares the performance of the blockchain based global attack detection paradigm with its BG counterpart with increasing number of regions under attack. We fix the maximum number of gossip rounds to 50 for each simulation case. We observe that the blockchain outperforms the gossip algorithm by a margin of approximately 20 simulation epochs. Figure 7.4 also indicates that the initial error for BG at each epoch decreases with an increase in the number of regions under attack. Such a scenario results in marginal improvement in the global attack detection.

As postulated by Theorem 7, low initial error is preferred for BG, which only happens when an increasing number of regions are under attack. On the other hand, a high number of gossip rounds are required for BG in order to overcome regional connectivity constraints predicted by Theorem 7. Both scenarios highlight the operational obstacles associated with an information diffusion scheme based on gossip. Figures 7.3, 7.4 conclusively show that the blockchain based framework delivers a reliable, timely and accurate detection as compared to its gossip counterpart in diverse operational scenarios.

## 7.7 Conclusion and Future Work

In this chapter, we present a decentralized blockchain based global attack detection mechanism that only uses locally reported alarm and its associated statistics to detect the onset of a global replay attack. We design a novel Bayesian update mechanism requiring one global multiplication and one global addition leading to a scalable and computationally efficient blockchain paradigm. We characterize the performance of the blockchain based global attack detection mechanism against a broadcast gossip based counterpart. In order to do so,

we first reformulate the computation of the global multiplication and addition operations to be amenable towards broadcast gossip. We then theoretically analyze the performance of the broadcast gossip with a limited precision blockchain version. Our analysis predicts an overall computational superiority of the blockchain version as opposed to the broadcast gossip. We implement and evaluate the blockchain based approach on a private Ethereum network with the help of Solidity for orchestrating the Smart Contracts. Our experiments demonstrate the accuracy of the decentralized detection mechanism as well as its robustness to increasing number of regions. Moreover, our results also corroborate our theoretical claim of computational superiority over the state-of-the-art, decentralized broadcast gossip paradigm by a significant margin. For our future work we plan to focus on local attacks that could be mutually correlated. We also aim to investigate a blockchain driven approach to distinguish between multiple failure modes and an attack.

# CHAPTER 8
# CONCLUSION

The use of decentralized methods to solve large scale planning problems in power systems has numerous benefits. Such problems becomes difficult to solve with increasing network size. Any approach that decomposes planning problems across time periods or scenarios cannot scale with increasing size of the network. For instance, if a method decomposes the problem in terms of scenarios, the subproblems still increase in size as the network becomes larger. The decentralized approach presented in this paper provides a region based geographic decomposition scheme that is scalable to increasing network size.

A decentralized communication model is more efficient in terms of communication latencies. A Master Slave communication model introduces a single point of failure with the master bearing the heavy burden of processing information sent by all the agents or workers. Since the master can only process information from one agent at a time, the other agents must wait till the master has had an opportunity to process and respond to their corresponding message. This results in poor scalability with respect to increasing agents or problem sizes. Since communication is more expensive than computation, idle time can be significantly reduced by allowing agents to send messages to each other instead of waiting on one master to respond. Significant reduction in idle time can be further achieved by adopting an asynchronous communication scheme among agents. As a result, asynchronous agent-to-agent communication improves computational performance compared to the agent-to-master communication model.

A decentralized computing model is applicable to a variety of computing platforms while promising good computational efficiency. They are ideal for a geographically dispersed computing setup like a network of cloud servers communicating over TCP/IP with high communication latencies. A Master Slave mechanism on the other hand falls under

the category of distributed computing models. A distributed computing model entails the master performing updates with information sent from workers. Such models are typically implemented and solved in a centralized but distributed environment like an HPC cluster with very low communication latencies. However, their applicability remains limited to such hardware systems only.

Decentralized computing models can provide privacy of data. The UC problem is a stepping stone towards more complex planning problems. Emerging applications of UC involving data-driven operations planning will not be amenable to a centralized model of computation for instance the large scale sensor driven maintenance problem. This will be particularly important with respect to new challenges in data acquisition and interpretations - for instance in the areas of integration of renewables, incorporation of maintenance, digital control, transmission line switching, prevention of cascading failures, among others. In such problems, co-locating data and problem components may prove to be infeasible making a decentralized mechanism more appealing.

In addition to exploring decentralized formulations, we explore two related paradigms in this research work as well. Our first exploration is in the direction of enabling a theoretical guarantee of privacy in decentralized mixed integer planning problems. While the decentralized formulations only guarantee data privacy on account of data isolation, we consider differential privacy as a means to ensure privacy with respect to the consensus flow variables as well.

Our second exploration relates to frameworks that enable a practical, real-world realization of a decentralized computational model. As a result, we consider the blockchain as a computational tool that provides the perfect platform for decentralized learning and aggregation among multiple parties. By utilizing Smart Contracts, the blockchain can enable decentralization of compute such that data privacy is preserved while eliminating the need for a central coordinator.

As part of the research presented so far we have explored the use of decentralized meth-

ods to solve large scale planning problems for power systems. We present two approaches for solving the decentralized UC problem in an asynchronous fashion. In the first, we apply local lagrangian updates towards flow and phase angle balance for iterative ADMM. It is observed that although significant computational benefit is detected, solution quality suffers on account of a Mixed Integer component in the problem. In the second approach, we make numerous fundamental changes to the previous version wherein we strengthen the convex phase using *Wei and Ozdaglar* aprroach to asynchronous constrained ADMM[33]. We employ a interleaved binary approach that makes our algorithm computationally more efficient and use a controller to satisfy theoretical assumptions made in [33]. We focus on the synchronous sensor driven maintenance model in our next work wherein we solve the yearly maintenance problem in a decentralized manner using sensor degradation signals. We show very good solution quality using the decentralized approach.

Our successive work consists of incorporating a commitment model in the operations of the maintenance problem. We use a hybrid shared and distributed memory approach to solve this problem efficiently. We decompose the network topology into regions and further decompose the regional local problem into weeks and employ OpenMP for parallelizing the local problem by weeks using the subgradient method. Our results show that the decentralized multithreaded, subgradient based approach can reduce solution times for large scale problems by a factor of up to 50 while delivering the same quality of results.

For demonstrating privacy guarantees on a mixed integer formulation, we leverage a differential privacy driven formulation for the short-term maintenance problem. We incorporate differential privacy concepts by creatively leveraging the theoretical properties of the Laplace distribution. We inject a well-calibrated, differentially private noise across the transmission lines between regions. We further incorporate control chart techniques in order to strengthen convergence and deliver a stable solution quality. Our work shows that differential privacy can be successfully applied to decentralized mixed integer formulations to protect flow values between regions while delivering a good solution quality.

In the last work presented in this research, we successfully apply blockchains and Smart Contracts to estimate the global cyber health status of a replay attack. Our approach delivers a faster way to detect global replay attacks compared to conventional decentralized schemes like Broadcast Gossip. Meanwhile, we show that our technique eliminates the requirement of a central coordinator while ensuring full data privacy.

The research presented in this document explores the varied themes relating to decentralization in power systems. The research themes span a broad spectrum ranging from novel formulations to solutions to practical, real-world problems as well as platforms to accomplish the same. We sincerely hope that our research paves the way forward for developing decentralized computational methodologies.

# Appendices

# APPENDIX A

## NOTATIONS FOR CHAPTER 2

| | |
|---|---|
| $\mathcal{G}$ | Set of all generators |
| $\mathcal{R}$ | Set of all regions |
| $\mathcal{N}_r$ | The neighboring regions of region $r$ |
| $\mathcal{N}_r^b$ | The neighboring regions of region $r$ having $b$ as foreign or boundary bus |
| $\mathcal{U}_r$ | The set of boundary buses |
| $\mathcal{V}_r$ | The set of all foreign buses in $r$ |
| $\mathcal{V}_r^b$ | The set of all foreign buses connected to bus $b$ |
| $\mathcal{I}_r$ | The set of internal buses |
| $\mathcal{B}_r$ | $\mathcal{U}_r \bigcup \mathcal{V}_r$ , the set of all boundary and foreign buses |
| $G_r$ | The set of all generators in region $r, \bigcup_{r=1}^{\lvert\mathcal{R}\rvert} G_r = \mathcal{G}$ |
| $G_{r,b}$ | The set of all generators connected to bus $b$ |
| $B^r$ | All internal and boundary busses of $r$ |
| $B_b^r$ | Neighbor buses of $b$ within region $r$ |
| $T$ | The planning horizon |
| $R_t^g$ | The replacement cost of generator $g$ |
| $\mu^g$ | The maintenance cost coefficient for generator $g$ |
| $M_t^g$ | The maintenance cost of generator $g$ $M_t^g = \mu^g R_t^g$, where $\mu^g$ is a constant |
| $D^g$ | The dispatch cost of generator $g$ |
| $C^g$ | The maximum capacity of generator $g$ |
| $y_t^g$ | The electricity dispatch of generator $g$ |

$d_t^b$ The demand at bus $b$

$\rho$ the penalty parameter of the problem, set in advance

# APPENDIX B

# NOTATIONS FOR CHAPTER 4

**Sets:**

| | |
|---|---|
| $\mathcal{R}$ | The set of all regions |
| $\mathcal{N}_r, G_r, \mathcal{U}_r, \mathcal{V}_r, \mathcal{I}_r$ | Neighboring regions, generators, boundary, foreign and internal buses of region $r$. |
| $\mathcal{B}_r$ | $\mathcal{U}_r \cup \mathcal{V}_r$, Boundary and foreign buses of $r$. |
| $\mathcal{N}_r^b, \mathcal{V}_r^b, G_r^b$ | Neighboring regions, foreign buses and generators connected to bus $b \in \mathcal{U}_r$ |
| $G_r^b, \mathcal{U}_r^b, \mathcal{V}_r^b, \mathcal{I}_r^b$ | Generators, boundary, foreign and internal buses connected to bus $b \in \mathcal{U}_r \cup \mathcal{I}_r$ |
| $\mathcal{B}_r^b$ | $\mathcal{U}_r^b \cup \mathcal{V}_r^b \cup \mathcal{I}_r^b$, Neighboring buses of bus $b$. |
| $T$ | Operations planning horizon |

**Decision Variables:**

Denoted for planning epoch $t \in T$

| | |
|---|---|
| $y_t^g$ | The electricity dispatch of generator $g$ |
| $x_t^g \in \{0, 1\}$ | The decision variable of generator $g$ |
| $\theta_t^b$ | The phase angle at bus $b$ |
| $\tilde{\theta}_t^{b,r'}$ | The phase angle of bus $b$ where $b \in \mathcal{U}_{r'}$ and $r' \in \mathcal{N}_r$ |
| $f_t^{uv}$ | Power flow from bus $u$ to $v$ such that $u \in \mathcal{U}_r$ and $v \in \mathcal{V}_r^u$ |
| $\pi_{Ut}^g, \pi_{Dt}^g$ | The up and down variable of generator g |

| | |
|---|---|
| $p_{r,t}$ | Production difference at region $r$ |
| $\lambda_t^b$ | The Lagrangian multiplier with respect to phase angles of bus b where $b \in \mathcal{U}_r \bigcup \mathcal{V}_r$ |
| $\phi_t^{uv}$ | The Lagrangian multiplier with respect to flow from bus $u$ to bus $v$ where $u \in \mathcal{U}_r$ and $v \in \mathcal{V}_r$ for any region $r$ |
| $\eta_t$ | The Lagrangian multiplier with respect to production difference of region $r \in \mathcal{R}$ |

**Constants:**

| | |
|---|---|
| $d^g, c^g, S_U^g, S_D^g$ | The dispatch cost, commitment cost start-up cost shut-down cost of generator $g$ |
| $P_{min}^g, P_{max}^g$ | Minimum and maximum capacity of $g$ |
| $M_U^g, M_D^g, R^g$ | Minimum up time, down time and ramp-up ramp-down constant for $g$ |
| $\delta_t^b$ | The demand at bus $b$ |
| $F_{max}^{uv}$ | Maximum capacity of line connecting buses $u$ and $v$ such that $u \in \mathcal{U}_r$ and $v \in \mathcal{V}_r^u$ |
| $\rho_\theta, \rho_f, \rho_p$ | Penalty parameter for phase angles, flows and production difference |
| $\Gamma^{uv}$ | Phase angle conversion for line $uv$ |

# APPENDIX C

# NOTATIONS FOR CHAPTER 5 AND CHAPTER 6

**Sets:**

| | |
|---|---|
| $\mathcal{R}$ | The set of all regions |
| $\mathcal{N}_r, G_r, \mathcal{U}_r, \mathcal{V}_r, \mathcal{I}_r$ | Neighboring regions, generators, boundary, foreign and internal buses of region $r$. |
| $\mathcal{B}_r$ | $\mathcal{U}_r \cup \mathcal{V}_r$, Boundary and foreign buses of $r$. |
| $\mathcal{N}_r^b, \mathcal{V}_r^b, G_r^b$ | Neighboring regions, foreign buses and generators connected to bus $b \in \mathcal{U}_r$ |
| $G_r^b, \mathcal{U}_r^b, \mathcal{V}_r^b, \mathcal{I}_r^b$ | Generators, boundary, foreign and internal buses connected to bus $b \in \mathcal{U}_r \cup \mathcal{I}_r$ |
| $\mathcal{B}_r^b$ | $\mathcal{U}_r^b \cup \mathcal{V}_r^b \cup \mathcal{I}_r^b$, Neighboring buses of bus $b$. |
| $T$ | Operations planning horizon |
| $M$ | Maintenance planning horizon |

**Decision Variables:**

Denoted for planning epoch $t \in T$

| | |
|---|---|
| $y_t^g$ | The electricity dispatch of generator $g$ |
| $x_t^g \in \{0, 1\}$ | The decision variable of generator $g$ |
| $\theta_t^b$ | The phase angle at bus $b$ |
| $\tilde{\theta}_t^{b,r'}$ | The phase angle of bus $b$ where $b \in \mathcal{U}_{r'}$ and $r' \in \mathcal{N}_r$ |
| $f_t^{uv}$ | Power flow from bus $u$ to $v$ such that $u \in \mathcal{U}_r$ and $v \in \mathcal{V}_r^u$ |

| | |
|---|---|
| $\pi^g_{Ut}, \pi^g_{Dt}$ | The up and down variable of generator g |
| $p_{r,t}$ | Production difference at region $r$ |
| $\lambda^b_t$ | The Lagrangian multiplier with respect to phase angles of bus b where $b \in \mathcal{U}_r \bigcup \mathcal{V}_r$ |
| $\phi^{uv}_t$ | The Lagrangian multiplier with respect to flow from bus $u$ to bus $v$ where $u \in \mathcal{U}_r$ and $v \in \mathcal{V}_r$ for any region $r$ |
| $\eta_t$ | The Lagrangian multiplier with respect to production difference of region $r \in \mathcal{R}$ |

**Constants:**

| | |
|---|---|
| $d^g, c^g, S^g_U, S^g_D$ | The dispatch cost, commitment cost start-up cost shut-down cost of generator $g$ |
| $P^g_{min}, P^g_{max}$ | Minimum and maximum capacity of $g$ |
| $M^g_U, M^g_D, R^g$ | Minimum up time, down time and ramp-up ramp-down constant for $g$ |
| $\delta^b_t$ | The demand at bus $b$ |
| $F^{uv}_{max}$ | Maximum capacity of line connecting buses $u$ and $v$ such that $u \in \mathcal{U}_r$ and $v \in \mathcal{V}^u_r$ |
| $\rho_\theta, \rho_f, \rho_p$ | Penalty parameter for phase angles, flows and production difference |
| $\Gamma^{uv}$ | Phase angle conversion for line $uv$ |

## REFERENCES


[1] D. Apostolopoulou, S. Bahramirad, and A. Khodaei, "The interface of power: Moving toward distribution system operators," *IEEE Power and Energy Magazine*, vol. 14, no. 3, pp. 46–51, 2016.

[2] S. K. Khaitan and J. D. McCalley, "Cyber physical system approach for design of power grids: A survey," in *2013 IEEE Power Energy Society General Meeting*, 2013, pp. 1–5.

[3] M. Yildirim, X. A. Sun, and N. Z. Gebraeel, "Sensor-driven condition-based generator maintenance scheduling part i: Maintenance problem," *IEEE Transactions on Power Systems*, vol. PP, no. 99, pp. 1–10, 2016.

[4] K. Hwang, J. Dongarra, and G. C. Fox, *Distributed and cloud computing: from parallel processing to the internet of things*. Morgan Kaufmann, 2013.

[5] C. Dwork, A. Roth, *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[6] M. Yildirim, X. A. Sun, and N. Z. Gebraeel, "Sensor-driven condition-based generator maintenance scheduling part ii: Incorporating operations," *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 4263–4271, 2016.

[7] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, ser. IPSN '04, Berkeley, California, USA: ACM, 2004, pp. 20–27, ISBN: 1-58113-846-6.

[8] J. Endrenyi, S. Aboresheid, R. N. Allan, G. J. Anders, S. Asgarpoor, R. Billinton, N. Chowdhury, E. N. Dialynas, M. Fipper, R. H. Fletcher, C. Grigg, J. McCalley, S. Meliopoulos, T. C. Mielnik, P. Nitu, N. Rau, N. D. Reppen, L. Salvaderi, A. Schneider, and C. Singh, "The present status of maintenance strategies and the impact of maintenance on reliability," *IEEE Transactions on Power Systems*, vol. 16, no. 4, pp. 638–646, 2001.

[9] N. P. Padhy, "Unit commitment-a bibliographical survey," *IEEE Transactions on Power Systems*, vol. 19, no. 2, pp. 1196–1205, 2004.

[10] N. Z. GEBRAEEL, M. A. LAWLEY, R. LI, and J. K. RYAN, "Residual-life distributions from component degradation signals: A bayesian approach," *IIE Transactions*, vol. 37, no. 6, pp. 543–557, 2005. eprint: http://dx.doi.org/10.1080/07408170590929018.

[11] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015. eprint: http://dx.doi.org/10.1137/14096668X.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[13] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the admm in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.

[14] Y. Wang, W. Yin, and J. Zeng, "Global convergence of admm in nonconvex nonsmooth optimization," *arXiv preprint arXiv:1511.06324*, 2015.

[15] A. Papavasiliou, S. S. Oren, and B. Rountree, "Applying high performance computing to transmission-constrained stochastic unit commitment for renewable energy integration," *IEEE Transactions on Power Systems*, vol. 30, no. 3, pp. 1109–1120, 2015.

[16] P. Srikantha and D. Kundur, "Distributed optimization of dispatch in sustainable generation systems via dual decomposition," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2501–2509, 2015.

[17] I. Aravena and A. Papavasiliou, "A distributed asynchronous algorithm for the two-stage stochastic unit commitment problem," in *2015 IEEE Power Energy Society General Meeting*, 2015, pp. 1–5.

[18] M. J. Feizollahi, M. Costley, S. Ahmed, and S. Grijalva, "Large-scale decentralized unit commitment," *International Journal of Electrical Power & Energy Systems*, vol. 73, pp. 97 –106, 2015.

[19] L. Dalcin, R. Paz, M. Storti, and J. D., "Mpi for python: Performance improvements and mpi-2 extensions," *Journal of Parallel and Distributed Computing*, vol. 68, no. 5, pp. 655 –662, 2008.

[20] G. O. Inc., *Gurobi optimizer reference manual*, 2015.

[21] *Pypower*, https://pypi.python.org/pypi/PYPOWER.

[22] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.

[23] O. Babaoglu, R. Davoli, L. A. Giachini, and M. G. Baker, "Relacs: A communications infrastructure for constructing reliable applications in large-scale distributed systems," in *Proceedings of the Twenty-Eighth Annual Hawaii International Conference on System Sciences*, vol. 2, 1995, pp. 612–621.

[24] T. Wu, K. Yuan, Q. Ling, W. Yin, and A. H. Sayed, "Decentralized consensus optimization with asynchrony and delays," *arXiv preprint arXiv:1612.00150*, 2016.

[25] R. Zhang and J. T. Kwok, "Asynchronous distributed admm for consensus optimization," in *Proceedings of the 31st International Conference on International Conference on Machine Learning*, ser. ICML'14, vol. 32, Beijing, China, 2014, pp. 1701–1709.

[26] K. Hedman, M. Ferris, R. O'Neill, E. Fisher, and S. Oren, "Co-optimization of generation unit commitment and transmission switching with n-1 reliability," in *IEEE PES General Meeting*, 2010, pp. 1–1.

[27] I Aravena and A Papavasiliou, "A distributed asynchronous algorithm for the two-stage stochastic unit commitment problem," *2015 IEEE Power Energy Society General Meeting*, pp. 1–5, 2015.

[28] V. M. Z. Kibaek Kim Cosmin G Petra, *An asynchronous bundle-trust-region method for dual decomposition of stochastic mixed-integer programming*, http://www.mcs.anl.gov/~kibaekkim/AsyncDD-preprint.pdf.

[29] V. M. Z. Kibaek Kim Mihai Anitescu, *An asynchronous decomposition algorithm for security constrained unit commitment under contingency events*, http://www.mcs.anl.gov/~kibaekkim/PSCC-KimAnitescuZavala.pdf.

[30] M. Assran and M. Rabbat, "An empirical comparison of multi-agent optimization algorithms," in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2017, pp. 573–577.

[31] A. Aytekin, "Asynchronous algorithms for large-scale optimization: Analysis and implementation," PhD thesis, KTH Royal Institute of Technology, 2017.

[32] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.

[33] E. Wei and A. Ozdaglar, "On the o (1= k) convergence of asynchronous distributed alternating direction method of multipliers," in *Global conference on signal and information processing (GlobalSIP), 2013 IEEE*, IEEE, 2013, pp. 551–554.

[34] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE transactions on automatic control*, vol. 31, no. 9, pp. 803–812, 1986.

[35] J. Liu, S. J. Wright, C. Ré, V. Bittorf, and S. Sridhar, "An asynchronous parallel stochastic coordinate descent algorithm," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 285–322, 2015.

[36] M. G. Rabbat and K. I. Tsianos, "Asynchronous decentralized optimization in heterogeneous systems," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, IEEE, 2014, pp. 1125–1130.

[37] T.-H. Chang, "A proximal dual consensus admm method for multi-agent constrained optimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3719–3734, 2016.

[38] J. Eckstein, "A simplified form of block-iterative operator splitting and an asynchronous algorithm resembling the multi-block alternating direction method of multipliers," *Journal of Optimization Theory and Applications*, vol. 173, no. 1, pp. 155–182, 2017.

[39] P. Ramanan, M. Yildirim, E. Chow, and N. Gebraeel, "Asynchronous decentralized framework for unit commitment in power systems," *Procedia Computer Science*, vol. 108, pp. 665–674, 2017.

[40] Y. Wang, L. Wu, and J. Li, "A fully distributed asynchronous approach for multi-area coordinated network-constrained unit commitment," *Optimization and Engineering*, pp. 1–34, 2018.

[41] J. Guo, G. Hug, and O. Tonguz, "Asynchronous admm for distributed non-convex optimization in power systems," *arXiv preprint arXiv:1710.08938*, 2017.

[42] H. Kopetz and W. Ochsenreiter, "Clock synchronization in distributed real-time systems," *IEEE Transactions on Computers*, vol. 100, no. 8, pp. 933–940, 1987.

[43] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, "Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning," in *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, IEEE, 2012, pp. 1543–1550.

[44] J. Ostrowski, M. F. Anjos, and A. Vannelli, "Tight mixed integer linear programming formulations for the unit commitment problem," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 39–46, 2012.

[45] R. Billinton, "Composite system maintenance coordination in a deregulated environment," *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 485–492, 2005.

[46] C. Feng and X. Wang, "A competitive mechanism of unit maintenance scheduling in a deregulated environment," *IEEE transactions on power systems*, vol. 25, no. 1, pp. 351–359, 2010.

[47] M. Marwali and S. Shahidehpour, "Integrated generation and transmission maintenance scheduling with network constraints," *IEEE Transactions on Power Systems*, vol. 13, no. 3, pp. 1063–1068, 1998.

[48] Y. Fu, M. Shahidehpour, and Z. Li, "Security-constrained optimal coordination of generation and transmission maintenance outage scheduling," *IEEE Transactions on Power Systems*, vol. 22, no. 3, pp. 1302–1313, 2007.

[49] Y. Fu, Z. Li, M. Shahidehpour, T. Zheng, and E. Litvinov, "Coordination of midterm outage scheduling with short-term security-constrained unit commitment," *IEEE Transactions on Power Systems*, vol. 24, no. 4, pp. 1818–1830, 2009.

[50] P. Ramanan, M. Yildirim, E. Chow, and N. Gebraeel, "An asynchronous, decentralized solution framework for the large scale unit commitment problem," *IEEE Transactions on Power Systems*, vol. 34, no. 5, pp. 3677–3686, 2019.

[51] Á. S. Xavier, F. Qiu, and S. S. Dey, "Decomposable formulation of transmission constraints for decentralized power systems optimization," *arXiv preprint arXiv:2001.07771*, 2020.

[52] M. L. Fisher, "The lagrangian relaxation method for solving integer programming problems," *Management science*, vol. 50, no. 12_supplement, pp. 1861–1871, 2004.

[53] A. Abiri-Jahromi, M. Fotuhi-Firuzabad, and M. Parvania, "Optimized midterm preventive maintenance outage scheduling of thermal generating units," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1354–1365, 2012.

[54] A. H. Elwany and N. Z. Gebraeel, "Sensor-driven prognostic models for equipment replacement and spare parts inventory," *IIE Transactions*, vol. 40, no. 7, pp. 629–639, 2008.

[55] A. Sugarman, "Condition monitoring of electrical equipment in nuclear power plants," *IEEE Transactions on Energy Conversion*, vol. EC-1, no. 3, pp. 1–8, 1986.

[56] T. Harris, *Rolling bearing analysis*. John Wiley and Sons, Inc.(United States), 1991.

[57] B. H. Chowdhury and S. Rahman, "A review of recent advances in economic dispatch," *IEEE transactions on power systems*, vol. 5, no. 4, pp. 1248–1259, 1990.

[58] F. Capitanescu, J. M. Ramos, P. Panciatici, D. Kirschen, A. M. Marcolini, L. Platbrood, and L. Wehenkel, "State-of-the-art, challenges, and future trends in security constrained optimal power flow," *Electric Power Systems Research*, vol. 81, no. 8, pp. 1731–1741, 2011.

[59] S. Yang, S. Tan, and J.-X. Xu, "Consensus based approach for economic dispatch problem in a smart grid," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4416–4426, 2013.

[60] J. Cortés, G. E. Dullerud, S. Han, J. Le Ny, S. Mitra, and G. J. Pappas, "Differential privacy in control and network systems," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, IEEE, 2016, pp. 4252–4272.

[61] Y. Wang, Z. Li, M. Shahidehpour, L. Wu, C. Guo, and B. Zhu, "Stochastic coptimization of midterm and short-term maintenance outage scheduling considering covariates in power systems," *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 4795–4805, 2016.

[62] J. Liu, H. Cheng, P. Zeng, L. Yao, C. Shang, and Y. Tian, "Decentralized stochastic optimization based planning of integrated transmission and distribution networks with distributed generation penetration," *Applied Energy*, vol. 220, pp. 800–813, 2018.

[63] M. F. Ghazvini, H. Morais, and Z. Vale, "Coordination between mid-term maintenance outage decisions and short-term security-constrained scheduling in smart distribution systems," *Applied energy*, vol. 96, pp. 281–291, 2012.

[64] C.-C. Sun, A. Hahn, and C.-C. Liu, "Cyber security of a power grid: State-of-the-art," *International Journal of Electrical Power & Energy Systems*, vol. 99, pp. 45–56, 2018.

[65] Y. Zhang, L. Wang, Y. Xiang, and C.-W. Ten, "Inclusion of scada cyber vulnerability in power system reliability assessment considering optimal resources allocation," *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 4379–4394, 2016.

[66] F. Zhou, J. Anderson, and S. H. Low, "Differential privacy of aggregated dc optimal power flow data," in *2019 American Control Conference (ACC)*, IEEE, 2019, pp. 1307–1314.

[67] F. Fioretto and P. Van Hentenryck, "Constrained-based differential privacy: Releasing optimal power flow benchmarks privately," in *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, Springer, 2018, pp. 215–231.

[68] F. Fioretto, T. W. Mak, and P. Van Hentenryck, "Differential privacy for power grid obfuscation," *IEEE Transactions on Smart Grid*, 2019.

[69] T. W. Mak, F. Fioretto, L. Shi, and P. Van Hentenryck, "Privacy-preserving power system obfuscation: A bilevel optimization approach," *IEEE Transactions on Power Systems*, 2019.

[70] F. Fioretto, T. W. Mak, and P. Van Hentenryck, "Privacy-preserving obfuscation of critical infrastructure networks," *arXiv preprint arXiv:1905.09778*, 2019.

[71] T. W. Mak, F. Fioretto, and P. Van Hentenryck, "Privacy-preserving obfuscation for distributed power systems," *arXiv preprint arXiv:1910.04250*, 2019.

[72] S. Kotz, T. Kozubowski, and K. Podgorski, *The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance*. Springer Science & Business Media, 2012.

[73] S. Mhanna, G. Verbič, and A. C. Chapman, "Adaptive admm for distributed ac optimal power flow," *IEEE Transactions on Power Systems*, vol. 34, no. 3, pp. 2025–2035, 2018.

[74] M. Rosas-Casals, S. Valverde, and R. V. Solé, "Topological vulnerability of the european power grid under errors and attacks," *International Journal of Bifurcation and Chaos*, vol. 17, no. 07, pp. 2465–2475, 2007.

[75] R. Kinney, P. Crucitti, R. Albert, and V. Latora, "Modeling cascading failures in the north american power grid," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 46, no. 1, pp. 101–107, 2005.

[76] R. L. Krutz, *Securing SCADA Systems*. Hungry Minds Inc, 2005, ISBN: 0764597876.

[77] D. Kushner, "The real story of stuxnet," *ieee Spectrum*, vol. 3, no. 50, pp. 48–53, 2013.

[78] J. Weiss, "Aurora generator test," *Handbook of SCADA/Control Systems Security*, p. 107, 2016.

[79] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 1, p. 13, 2011.

[80] Y. Mo, R. Chabukswar, and B. Sinopoli, "Detecting integrity attacks on scada systems," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1396–1407, 2014.

[81] R. S. Smith, "A decoupled feedback structure for covertly appropriating networked control systems," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 90–95, 2011.

[82] Y. Benjamini and D. Yekutieli, "The control of the false discovery rate in multiple testing under dependency," *Annals of statistics*, pp. 1165–1188, 2001.

[83] *Multiyear plan for energy sector cybersecurity*, https://www.energy.gov/sites/prod/files/2018/05/f51/DOE%20Multiyear%20Plan%20for%20Energy%20Sector%20Cybersecurity%20_0.pdf.

[84] G. Fisk, C. Ardi, N. Pickett, J. Heidemann, M. Fisk, and C. Papadopoulos, "Privacy principles for sharing cyber security data," in *2015 IEEE Security and Privacy Workshops*, IEEE, 2015, pp. 193–197.

[85] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE transactions on information theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[86] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Transactions on Signal processing*, vol. 57, no. 7, pp. 2748–2761, 2009.

[87] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 645–658, 2011.

[88] M. Caselli, E. Zambon, J. Amann, R. Sommer, and F. Kargl, "Specification mining for intrusion detection in networked control systems.," in *USENIX Security Symposium*, 2016, pp. 791–806.

[89] D. I. Urbina, J. A. Giraldo, A. A. Cardenas, N. O. Tippenhauer, J. Valente, M. Faisal, J. Ruths, R. Candell, and H. Sandberg, "Limiting the impact of stealthy attacks on industrial control systems," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2016, pp. 1092–1105.

[90] D.-H. Kang, B.-K. Kim, and J.-C. Na, "Cyber threats and defence approaches in scada systems," in *16th International Conference on Advanced Communication Technology*, IEEE, 2014, pp. 324–327.

[91] M. S. Rahman, M. A. Mahmud, A. M. T. Oo, and H. R. Pota, "Multi-agent approach for enhancing security of protection schemes in cyber-physical energy systems," *IEEE transactions on industrial informatics*, vol. 13, no. 2, pp. 436–447, 2017.

[92] T. Huang, B. Satchidanandan, P. Kumar, and L. Xie, "An online detection framework for cyber attacks on automatic generation control," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6816–6827, 2018.

[93] J. Wan, J. Li, M. Imran, D. Li, and Fazal-e-Amin, "A blockchain-based solution for enhancing security and privacy in smart factory," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3652–3660, 2019.

[94] K. Gai, Y. Wu, L. Zhu, M. Qiu, and M. Shen, "Privacy-preserving energy trading using consortium blockchain in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3548–3558, 2019.

[95] M. Mylrea and S. N. G. Gourisetti, "Blockchain: A path to grid modernization and cyber resiliency," in *2017 North American Power Symposium (NAPS)*, 2017, pp. 1–5.

[96] Y. Wang, Z. Su, and N. Zhang, "Bsis: Blockchain-based secure incentive scheme for energy delivery in vehicular energy network," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3620–3631, 2019.

[97] M. L. Di Silvestre, P. Gallo, M. G. Ippolito, E. R. Sanseverino, and G. Zizzo, "A technical approach to the energy blockchain in microgrids," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4792–4803, 2018.

[98] G. Liang, S. R. Weller, F. Luo, J. Zhao, and Z. Y. Dong, "Distributed blockchain-based data protection framework for modern power systems against cyber attacks," *IEEE Transactions on Smart Grid*, 2018.

[99] N. Alexopoulos, E. Vasilomanolakis, N. R. Ivánkó, and M. Mühlhäuser, "Towards blockchain-based collaborative intrusion detection systems," in *International Conference on Critical Information Infrastructures Security*, Springer, 2017, pp. 107–118.

[100] B. D. Anderson and J. B. Moore, "Optimal filtering," *Englewood Cliffs*, vol. 21, pp. 22–95, 1979.

[101] M. Athans, "The role and use of the stochastic linear-quadratic-gaussian problem in control system design," *IEEE transactions on automatic control*, vol. 16, no. 6, pp. 529–552, 1971.

[102] K. P. Dan Li Nagi Gebraeel, "Detection and differentiation of replay cyberattack and equipment faults in scada systems," 2019, , Submitted to: IEEE Transactions on Automation Science and Engineering.

[103] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "Pbft vs proof-of-authority: Applying the cap theorem to permissioned blockchain," 2018.

# VITA

Paritosh Ramanan is PhD Candidate in Computational Science and Engineering with the School of Industrial and Systems Engineering at Georgia Institute of Technology in Atlanta, Georgia. His research focuses on developing decentralized algorithms for improved computational performance of large scale optimization problems through the use of parallel and distributed computing paradigms. In 2018, he was selected as a Sam Nunn Security Fellow, where he worked on policy inspired research on improving the cybersecurity of the national power grid. During his Ph.D. he spent two amazing summers as a research intern at Sandia National Labs in Albuquerque, New Mexico as well as NEC Research Labs in San Jose, California respectively.

Prior to his PhD he earned a Masters in Computer Science from Georgia State University in Atlanta, Georgia in 2015. As part of his Masters thesis research, he was involved in designing and building a distributed computational platform for seismic tomography. He participated in a 10 day expedition to implement his platform on the Llaima Volcano in Southern Chile.

He obtained his Bachelors in Information Systems from Birla Institute of Technology and Science (BITS) Pilani, Goa Campus in 2013. His research during his undergrad years, focused on techniques meant to establish connectivity in wireless sensor networks.